

AD-A124 791

STRUCTURAL MODEL TUNING VIA VECTOR OPTIMIZATION(U) AIR
FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF
ENGINEERING C R DEVORE DEC 82 AFIT/GAE/AA/82D-8

1/1

UNCLASSIFIED

F/G 13/13

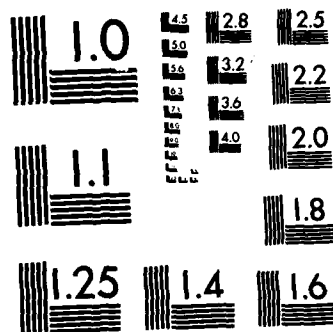
NL

END

FILMED

14

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ADA 124 791

AIR FORCE INSTITUTE OF TECHNOLOGY



AIR UNIVERSITY
UNITED STATES AIR FORCE

STRUCTURAL MODEL TUNING

VIA VECTOR OPTIMIZATION

THESIS

AFIT GAF MC 820-8 CHARLES R. DEVORE
CAPT USAF

SCHOOL OF ENGINEERING

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

This document has been approved
for public release and sale; its
distribution is unlimited.

DTIC
SELECTED
FEB 23 1982

E

DTIC FILE COPY

88 02 0221

1

STRUCTURAL MODEL TUNING
VIA VECTOR OPTIMIZATION

THESIS

AFIT/GAE/AA/82D-8

CHARLES R. DEVORE
CAPT USAF

OTIC
1

Approved for public release; distribution unlimited

STRUCTURAL MODEL TUNING
VIA VECTOR OPTIMIZATION

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Charles R. DeVore, B.S.A.A.E.

Capt

USAF

Graduate Aerospace Engineering

December 1982

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	



Approved for public release; distribution unlimited

Preface

I thank my two indispensable advisors, Capt Hugh C. Briggs and Capt Aaron R. DeWispelare, who were able to give me guidance and suggestions to help keep me progressing on the right path. I also thank Maj M. M. Wallace for taking the time to participate as a member of my thesis committee.

My thanks go to my fellow classmate, Capt George G. London, Jr., who gave me advice on how to use NASTRAN and its output and provided technical assistance on many aspects of the T-38 stabilator example.

I also want to give special thanks to my wife, Kathy, for her support throughout this entire 18 month graduate program. Her typing skills and extra encouragement during the manuscript preparation time were especially appreciated.

Charles R. DeVore

Contents

	<u>Page</u>
Preface	ii
List of Figures	iv
List of Tables.	v
Abstract.	vi
I. INTRODUCTION	1
Purpose.	1
Background	2
Problem.	5
Approach to the Problem.	6
II. TUNING PROCEDURE	7
Introduction	7
The MOOT Process	7
Objective Functions.	11
The NDSS and the Decision Maker.	14
Applications	18
III. FINITE ELEMENT MODEL EXAMPLE	21
Introduction	21
History.	21
Objective Functions.	27
Decision Variables	31
Constraints.	32
Finite Element Model Results	33
Extension of Example	38
IV. CONCLUSIONS AND RECOMMENDATIONS.	41
Bibliography.	43
APPENDIX A: The T-38 Stabilator.	45
APPENDIX B: Displacement and Frequency Data for T-38 Stabilator.	49
APPENDIX C: Data Structure in PROCES	54
APPENDIX D: Optimization Data of T-38 Stabilator	63
Vita.	71

List of Figures

<u>Figure</u>		<u>Page</u>
1	T-38 Stabilator	47
2	Finite Element Model.	48
3	PROCES Flow Diagram	57
4	Data Structure in PROCES.	59
5	Subroutine RESTNT	60
6	Variable Space for 202 Member NDSS.	67
7	2-D View of Variable Space for 202 Member NDSS.	68
8	Variable Space for 255 Member NDSS.	69

List of Tables

<u>Table</u>		<u>Page</u>
I	Measured and Calculated Frequencies	26
II	Static and Frequency Decision Variable Values	50
III	Total Static Load Case Errors	51
IV	Static Load Case 2 Errors	51
V	Static Load Case 4 Errors	52
VI	Static Load Case 10 Errors.	52
VII	Total Frequency Errors.	53
VIII	First Torsional Mode Frequency Errors	53
IX	Best Models for T-38 Stabilator	66
X	Flutter Analysis Data	70

Abstract

This report examines tuning a finite element model using vector optimization techniques. Structural models using finite element theory often need to be adjusted so they can accurately simulate the real structure. The goal is to tune the model such that it will reproduce data derived from the structure.

The tuning procedure is complicated by many factors. First of all, the model may be required to reproduce several sets of different performance data which may have conflicting effects on the model. The number and kinds of parameters to modify may be large or uncertain. Also, there may be many constraints on the model. With all these considerations and several performance indices to extremize, there may be many compromise solutions to examine in order to select the best one.

Therefore, the problem has been broken into two parts. First, the performance indices are extremized using multiple objective optimization theory, producing a set of possible solutions. Next, the solutions are rank ordered according to a decision maker's preferences. The solution ranked number one is, then, the best answer.

The tuning process was applied to a T-38 horizontal stabilator. Four static deformation and two natural frequency experimentally determined data sets were used as the objective functions for the three parameter model. This data was compared to the analytical data computed by NASTRAN, producing a set of over 200 models as the solution for the constrained, non-linear problem. Numerous weighting combinations indicated that only five of the solutions were of

interest. These five solutions contained a best static deformation model, a best frequency model and three intermediate combinations of these two models.

The automated procedure outlined in this report proved to be a versatile method capable of solving many types of tuning problems.

STRUCTURAL MODEL TUNING

VIA VECTOR OPTIMIZATION

I Introduction

Purpose

Structural models using finite element theory often need to be tuned, or adjusted, before they can accurately simulate the real structure. Tuning is necessary since the model may not behave to loading conditions in the same manner as the real structure. This difference occurs because the structure's members are modeled by less accurate finite elements which can closely approximate, but not exactly duplicate, all of the member's physical characteristics and boundary conditions. The model can be improved by having its physical and material properties adjusted until its performance duplicates the multiple types of structural performance. Each of these performances can be in any one of several different forms such as displacements, natural frequencies or mode shapes. Diverse performances such as these may depend upon the model differently, possibly requiring the model to have conflicting characteristics. Decreasing the difference between the model's performance and the structure's actual performance is the desired goal, or objective, of the tuning process. A procedure to accomplish this tuning process for these multiple objectives will be presented in this report.

Background

A finite element model is composed of a few simple types of elements combined in a particular number and order in an attempt to represent the real structure as accurately as possible. The elements are termed simple because they are simpler representations of their physical counterparts and cannot exactly duplicate the complex behavior of the actual structural members. Also, the model may be a less complicated arrangement of pieces than the hardware in order to make the analysis easier. Therefore, the model will probably not have exactly the same physical capabilities as the structure.

The finite element model can be improved by either increasing the number of nodes or adjusting the model's parameters. Increasing the number of nodes will add more elements and joints, changing the model's ability to deform. More importantly, though, more nodes increases the number of equations to be solved in order to obtain the model's displacements, leading to increased computer run time (i.e., cost). As mentioned earlier, the elements used to represent the structural components are simpler idealizations of those components which will not exactly duplicate the component in bending and torsion if loaded identically. Since the finite elements are not exact duplicates of the real components, they need not necessarily have the same physical and material properties. Increasing the number of nodes, then, may not improve how an element duplicates a component, but altering the element's properties may. Therefore, the properties can be adjusted to obtain the best simulation of the component. However, the total number of parameters to be adjusted for all the elements of a large model could easily be prohibitive. The few parameters most effecting the

desired performances, then, must be determined to keep the variable set small.

The selected parameters are adjusted to force the model to duplicate a given set of characteristics which make up the objectives of the tuning process. These characteristics may be experimentally determined results from the actual structure, exact analytical solutions or any other characteristic data the analyst has confidence in. The characteristics for the finite element model will consist of performance data such as the displacements of the model at various nodes due to different loading conditions, the natural frequencies of the model or the mode shapes for each natural frequency. The performances may be in various forms. The displacements in units of length and the frequencies in cycles per second are examples. Some of these performances may also be more reliable than others and should be given greater emphasis in the tuning process.

Performances of different types such as these, which may also have very different magnitudes, are difficult to compare. Some type of normalizing procedure must be employed to non-dimensionalize them and put them on a comparable scale. The performance data may be in the form of continuous functions or tables of discrete data points with respect to the independent property parameters. They may require conflicting parameter values such as an abnormally large displacement requiring more stiffness in the model to decrease the displacement and at the same time a higher than desired natural frequency requiring less stiffness to reduce it. The model tuning process must account for all of these possibly conflicting concerns when it adjusts a parameter.

Each model is defined by a set of parameters (the decision

variable vector) and has a set of performances associated with it. This vector lives in a multi-dimensional space with each point in the space representing a different model. Likewise, the performance indices could be said to represent a performance space. The transformation of a point from the model space to the performance space may be difficult. For simplicity, the transformation may be thought of as a "function" which transforms the decision variables into a performance. Each performance requirement has a different "function." For many tuning (optimization) problems, the "function" is truly a single mathematical statement. Even some complex performance prediction processes can be modelled or curve fitted with a single mathematical statement. The transformation for the finite element model involves a finite element computer program with the relationship between the model and performances unclear. The program must be evoked each time a new variable set is to be analyzed and will yield a set of performances for the one decision variable vector. Further, more than one model, or variable set, may produce the same performance combination. The analyst or decision maker will then have to select from among the many possible models the one which will best fit his particular problem.

The desirability of a particular performance combination will dictate which model is best for the decision maker's situation. The finite element model will generally be able to imitate only a few of the performances accurately. A model can usually be found easily which duplicates one performance type very well, but multiple performance requirements may lead to many models each of which will do a different combination of performances well. Altering the model may tune it to

one combination of performances while forcing it away from the original set. The best model will be the one which duplicates that performance set most pertinent to the problem at hand. The other performances which were not considered may not be matched very well for that model. The decision maker, then, has to determine which performance characteristics he needs and then weight each performance to reflect its importance. The best model will be a compromise which may not be appropriate for a different situation.

Air Force analysts need to be able to tune finite element models to accurately predict the behavior of real structures. The finite element models for complex structures, such as aircraft wings, are often in error enough to justify the time and cost of tuning the model. The models are used to predict deformations due to loads such as stores or maneuvers and to predict aircraft flutter speeds. Flutter speeds can be degraded by repaired battle damage, accidents or deterioration. The changes to the flutter speeds must be known in order to set the maximum safe flight speed. Since these predictions may help establish aircraft performance or repair limitations or capabilities, they can save or cost the Air Force valuable resources. It is important, then, to have accurate finite element models.

The Problem

This thesis will investigate the use of optimization theory to tune a finite element model of a structure to accurately predict a desired combination of performances. Since there may be many tuned solutions for the multiple performances, the tuning process will be defined as well as a method for selecting the best solution from among all tuned candidates.

Approach to the Problem

A multiple objective optimization procedure and a performance weighted model selection scheme will be used to solve this problem. The goal of tuning a finite element model is to minimize the difference between the predicted and the measured deformations and natural frequencies. Various material and physical properties will make up the decision variables. These variables will be adjusted to make the desired deformation and frequency errors a minimum. Also, certain physical and material constraints will be observed during the tuning in order to maintain the credibility of the model. Multiple objective optimization theory (MOOT, Ref 1) has the capability to perform this tuning process and will be the major tool of this study. The performance errors for each model will be calculated by the MOOT process based upon stored information. Then, the decision variable vector will be tuned to minimize the performance errors. A preference for different combination of performance errors will generate different candidate solutions. A performance weighting method which reflects the preference the analyst places on the set of errors will then be employed to select the best solution (i.e., model) for his situation. The tuning process developed as outlined above will also be demonstrated on a structure of current interest to the Air Force.

II Tuning Procedure

Introduction

Multiple objective optimization theory can be used to locate the problem solution (the finite element model) which minimizes the desired combination of performance errors. As explained by DeWispelare in Ref 1, MOOT can be mechanized by implementing either the constraint or the weighting technique. Using a scalar optimization routine, either technique will generate the optimal solution for a discrete combination of performance errors. The solution is then compared to other solutions and is retained if it is not dominated by one of them. The retained solutions make up the non-dominated solution set. The performance error combination is altered and the procedure repeated. When the analyst is finished examining all possible combinations, the decision maker can apply his own preference criteria to choose which solution from the set is best for his situation.

The MOOT Process

The Problem. A multiple-criteria problem to be optimized by a MOOT procedure consists of selecting a decision variable vector such that a combination of objectives is extremized (either maximized or minimized) while satisfying various constraints. The problem is stated as follows:

$$\begin{aligned} &\text{extremize } \underline{Z}(\underline{X}) \\ &\text{subject to } g_i(\underline{X}) \leq 0 \quad ; \quad i=1,2,\dots,m \\ &\text{and } h_i(\underline{X}) = 0 \quad ; \quad i=m+1,\dots,m+mz \end{aligned}$$

where \underline{Z} is made up of all the performance indices for the problem which

are each a different function of \underline{X} . The m constraints $g_i(\underline{X})$ are the inequality constraints and the m_2 constraints $h_i(\underline{X})$ are the equality constraints. Constraints of $x_j \geq 0$, where j ranges from 1 to n , may be added to ensure positive decision variable values. Here, n is the total number of decision variables.

The optimization of a vector of performance indices (objective functions) cannot be accomplished mathematically in a generic fashion. A possible approach to extremizing a vector of performance indices is to repose the problem as a pseudo-scalar formulation which can then be solved iteratively (Ref 1). The weighting technique and the constraint technique are two methods to put the multiple objective problem into a scalar form for the optimization routine. The weighting technique extremizes a single function which is formed from a weighted sum of all the objectives. Investigation of the feasible space is accomplished by varying the set of weights. The constraint technique optimizes one performance index while maintaining the other objectives at prescribed values as constraints. Each prescribed set of objectives produces a new solution. The constraint technique was chosen for this problem since it can identify the solution set more accurately.

PROCES. A very versatile vector optimization routine which incorporates the constraint technique is a program called PROCES (see Appendix C and Ref 2). PROCES has been modified at the Air Force Institute of Technology to include various scalar optimization methods. The version of PROCES used in this effort contains the Sequential Unconstrained Minimization Technique (SUMT, Ref 3) as the scalar optimization routine. SUMT will adjust a decision variable vector to optimize a linear or non-linear objective function while attempting to

satisfy numerous constraints. SUMT contains options for the Newton-Raphson, steepest descent and McCormick modified Fletcher-Powell minimization techniques. These minimization techniques require particular conditions on the number of decision variables. In order to ensure that a minimum exists for the problem, there must be fewer decision variables than equality constraints. Also, the inequality constraints must each have one and only one minimum value (see Ref 3). The finite element problem has the sufficient conditions, then, since there are no equality constraints and all of the inequality constraints are convex. Any of the constraints which are not satisfied by the model are appended to the objective function, $Z_i(\underline{X})$, as penalties. SUMT then minimizes the objective function which has been augmented by the penalty terms. Maximization is accomplished by minimizing the negative of the objective function. The constrained problem is in the form of:

$$\begin{aligned}
 &\text{extremize } Z_i(\underline{X}) \\
 &\text{subject to } g_j(\underline{X}) \leq 0 \quad ; \quad j=1,2,\dots,m \\
 &\quad \quad \quad h_j(\underline{X}) = 0 \quad ; \quad j=m+1,\dots,m+m_2 \\
 &\quad \quad \quad x_k \geq 0 \quad ; \quad k=1,2,\dots,n \\
 &\text{and } L_1 < Z_1 < U_1 \quad ; \quad \text{for all } l \text{ with } l \neq i.
 \end{aligned}$$

L_1 and U_1 are the lower and upper bounds of each performance index Z_1 , where l and i vary from one up to the total number of objective functions. The lower and upper bounds can be established by such means as professional opinion or experimental or theoretical results. These bounds, then, depend upon the problem. Again, m is the total number of inequality constraints, m_2 is the number of equality constraints and n is the number of decision variables.

The vector optimization problem is transformed into the minimization of a single objective function for SUMT by treating all but one of the performance indices as constraints. The new problem formulation is as follows:

$$\begin{aligned}
 &\text{minimize} && Z_i(\underline{X}) \\
 &\text{subject to} && g_j(\underline{X}) \leq 0 \quad ; \quad j=1,2,\dots,m \\
 & && h_j(\underline{X}) = 0 \quad ; \quad j=m+1,\dots,m+m_2 \\
 & && x_k \geq 0 \quad ; \quad k=1,2,\dots,n \\
 &\text{and} && Z_r = Z_l \quad ; \quad \text{for all } l \neq i \text{ and } l=r \\
 &\text{where} && L_1 < Z_r < U_1.
 \end{aligned}$$

Now, $Z_i(\underline{X})$ represents only the i th objective function while l and r vary over all the other performance indices. The Z_r are prescribed values of the performance index which are iteratively changed between the bounds L_1 and U_1 such that all of the objective function space is examined.

The Constraints. The constraints on the finite element model problem are the structural constraints, such as pins and rollers, and the performance constraints, which consist of the prescribed values for the Z_r performance errors. The structure will have feasibility constraints which will limit the solution. For example, if a physical dimension such as thickness is being tuned, it will have some small positive value or zero as a lower limit and some larger value as an upper limit. The model will not be feasible beyond these limits due to construction, overall structural weight, aerodynamic or other applicable reason. Likewise, material properties such as Young's Modulus will have limits. Also, all the variables for a finite element model are usually required to be positive since they are all physical

or material variables which cannot realistically have negative values. The analyst can select which objective function to optimize while all the other indices are constrained within a judiciously chosen performance space. The solution must satisfy all of these constraints at least within some specified tolerance.

Local Minima. The three scalar minimization techniques available in PROCES are like other schemes in that they find only local minima (Ref 4). These schemes all use local information about the performance index being optimized to locate the minimum. The scheme may not move the \underline{X} far enough from some local minimum objective value to locate another minimum which may be a lower value. The technique may stop at the local minimum, then, instead of searching for a global minimum. One method to combat this deficiency is to begin the procedure at various starting points in order to locate the global minimum.

The ability to predict the presence and location of local minima a priori diminishes as the number of objective functions increases. This local minimum problem is not apparent for a finite element model with only one performance index. But, as multiple performances are examined, the surface of feasible performance error combinations in the multi-dimensional performance space becomes very complex.

Objective Functions

The Goal. The objective functions augmented with any constraint violations provide the measure of "goodness" for the \underline{X} which helps guide the solution process to the desired result. As the optimization procedure alters the \underline{X} , the amount of improvement in the $Z_i(\underline{X})$ and penalties indicate whether or not the optimization technique is

proceeding in the correct direction through the performance space to a minimum. The minimum is assumed when no more improvement can be obtained. The performance indices, then, strongly influence the answer. Changing the magnitude or shape of an objective function may lead the optimization procedure to a different solution. The goal for the finite element model is to make the measure of "goodness," or the difference between the predicted and actual structural performances, as small as possible, preferably zero.

The Objective Function Form. The objective functions can be in any of several different forms. The particular form is of no consequence to the optimization routine as long as a value exists for each \underline{X} and it can be numerically differentiated. When the optimization routine calls for an evaluation for the current \underline{X} , the evaluation method must produce the appropriate performance set. Every optimization routine will require the \underline{X} evaluation as it searches through the decision variable space no matter how sophisticated or simple the search method is. Each routine will expect the objective function's and constraints' values in return. The manner in which the values are produced is not specified by the search routine.

For instructional purposes in optimization courses, each performance index is generally a single mathematical statement (see Ref 4). This statement depends upon the variables X_i . Usually, perhaps through curve fitting to data, this form can also be used to represent almost any performance function. For example, an equation fitted to a finite number of experimental data points may be a good representation of the performance. In this case, the equation could be used as the objective function in place of the undetermined complicated function.

As the problem representation becomes more complex, the functional form may be a long series of complicated calculations, perhaps even an entire computer program. The performance of a finite element model is evaluated using a finite element computer program. If the program is small and fast enough, it could be implemented directly under the optimization routine. When an \underline{X} evaluation is necessary, the program could be called as a subroutine, computing and returning the performance and constraint values corresponding to the \underline{X} . In some cases, though, the program is too large or time consuming to be placed directly in the flow of the optimization routine. In such a case, the program could be exercised separately from the optimization routine to evaluate discrete combinations of the \underline{X} .

Tabular data can be used to simulate the structural analysis program. Prior to optimizing, the performance values could be obtained for discrete \underline{X} 's covering the decision variable space. The values could then be represented by a single statement curve fitted to the data or by a table of values. Either the equation or the table could be used in place of the program to evaluate the \underline{X} quickly. If a table of values is used, a table look-up procedure would be required to obtain the performance values for any \underline{X} in the decision variable space.

The finite element computer program used in this thesis is quite large and too time consuming to be used directly, so performance data will be incorporated in the form of tables. The appropriate performance values will be found with a table look-up routine using linear interpolation. Figures 4 and 5 in Appendix C show how the data is used in subroutine RESTNT within PROCES. The subroutine SPACINT called by RESTNT is the interpolation routine. Both of these

subroutines are contained in the SUMT block of Appendix C's Figure 3.

Solution Space Gaps. Gaps where no feasible solutions exist in the solution space may exist due to the diverse and possibly conflicting nature of the various kinds of performances and can stop the automated progression of the tuning process. An \underline{X} must exist for a particular combination of performances before that set can be feasible. It is possible that some of the constrained performance sets prescribed while methodically examining the performance space may not have a feasible \underline{X} . Void regions may exist for even a single objective function. For example, an objective function which is always positive will have no feasible solution with the performance index less than the value of zero. Requiring an \underline{X} to produce a negative performance would then be futile. Holes in the objective function space can become more troublesome and much more likely as the number of objective functions increases. Gaps in the solution space of the finite element model are possible where performances are attempting to drive the solution in conflicting directions. Of course, this is problem dependent and may or may not be of concern. If gaps do exist, the optimization routine will not be able to locate solutions in those regions. A different routine or parameter adjustments (i.e., tolerance value, iteration limits, etc.) may be required. The tuning procedure will have to be flexible enough to allow a solution along the edge of the void, where all the performance constraints will not be met very well. Thus, if voids do exist, bunching of the solutions along the edges of the performance gaps will have to be tolerated.

The Non-Dominated Solution Set and The Decision Maker

Each solution obtained through the vector optimization process is

a candidate for the non-dominated solution set. The best of all possible situations would have one solution for the problem. This one solution would extremize each performance index having each one greater than or equal to the corresponding index of any other solution with at least one performance index strictly greater than its counterpart. This case, however, seldom occurs. Most constrained problems will have particular performance indices which are extremized by one solution while a different grouping of performance indices is extremized by another solution. As some of the competing performance indices are improved, others are made worse. No solution is obviously better than another. Therefore, a large number of good solutions may exist, each one having at least one of its performance indices better than the corresponding one from another solution. The non-dominated solution (NDS) can be mathematically defined as:

$$\begin{aligned} \underline{X}' \text{ is a NDS iff } & \nexists (\underline{X}) \in \underline{X} \\ \text{such that } & Z_r(\underline{X}) > Z_r(\underline{X}') \text{ for some } r=1,2,\dots,n \\ \text{and } & Z_s(\underline{X}) \geq Z_s(\underline{X}') \quad \forall s \neq r \end{aligned}$$

where \underline{X} is the decision variable space. This states that a solution \underline{X}' is a NDS if and only if there does not exist a solution \underline{X} which is contained in the decision variable space such that a performance index $Z_r(\underline{X})$ is greater than $Z_r(\underline{X}')$ while all other performance indices $Z_s(\underline{X})$ are greater than or equal to $Z_s(\underline{X}')$. Each of the solutions passing this test is a member of the non-dominated solution set (NDSS) for the maximization case. For our finite element model minimization problem, the same definition can be used with the signs of the performance indices all changed.

In practice, the NDSS is obtained by examining each candidate solution for dominance. Each performance constraint combination in the performance grid produces a candidate solution for the NDSS. A candidate has each of its performance indices compared to the performance indices of each member of the NDSS in turn. If a NDSS member has each of its performance indices better than the candidate's, the candidate solution is discarded since it is dominated. If a candidate has even one of its performance indices better than the member's, it is retained for comparison with the next member. If it is retained through all NDSS member comparisons, it becomes a new member of the NDSS. The new member could also dominate some of the old members, in which case the old member would be discarded. If, however, the new and old members have the same performance indices, their X's must be examined to determine if two different model produced the same performances. If their X's are different, both are kept as NDS members; if their X's are the same, either the old or the new member is discarded. In this manner, only those solutions which are truly unique are kept. Figure 4 outlines this entire procedure.

Each solution in the NDSS is not differentiable as better in its vector form until some sort of preference scheme is applied to select the best solution for the situation. The decision maker decides how important each performance trait is for his situation and then weights each performance index according to its importance. In addition, the weighting may be increased for those performance values in which there is more confidence or which may be more pertinent to the given situation. The performance values are then multiplied by their

corresponding weights and the results summed as the score. The score obtained is a number indicating the amount of preference when compared to other scores. The NDSS can be rank ordered by comparing the scores for each NDSS member. Thus, the NDSS contains possible solutions of the given problem. A modification to the problem would change the NDSS; however, a modification of the weighting system may change the rank ordering but not the members of the NDSS. If the importance of any performance index is changed, the weighting system will change and, therefore, the final solution may change. Non-technical issues, then, as manifested in the analyst's performance weighting, will determine which NDSS member is the most appropriate for the situation.

The performance indices are usually different in ranges and magnitudes. Multiplying these values by the weights could apply a higher than desired influence on the performance indices with the larger values, biasing the scores in favor of these indices. This biasing is compounded when a performance index changes more rapidly than others as the X changes or has a wider range than other indices since this index would be more sensitive to the weights than other indices. These problems are eliminated by normalizing the performance indices with respect to their ranges and magnitudes and requiring the weights to be a percentage with the total equalling 100. The normalized performance is computed as follows:

$$Z_{\text{NORM},i} = \frac{Z_i - ZLB_i}{ZUB_i - ZLB_i} ; \text{ for all } i.$$

Here, ZLB_i and ZUB_i are the lower and upper bounds, respectively, for the performance index. The normalized performance indices are always positive and less than one and the weights are percentages which

will be between zero and plus one. Multiplying these two, then, and summing each value will produce a score which is always between zero and plus one. The normalized performance indices allow truly unbiased tradeoffs to be accomplished so the NDSS can be analyzed fairly.

This weighted selection procedure was used in this thesis in the form of a stand-alone computer program. The interactive program was derived from a version contained and demonstrated in the PROCES of Ref 2. The program was modified as necessary to contain the capabilities explained above. The program was designed to be used to rank order the NDSS in real time according to the analyst's (or decision maker's) preferences. The analyst could then change his preferences as the situation demanded or examine other possible preferences as he deemed necessary.

Applications

The vector optimization process described above has been developed for the finite element model, but with only a few modifications it can be adapted to other applications. As mentioned previously, the objective functions can be in many forms. This opens the door to almost any problem for which an optimum solution is desired. The problem can have one or more objective functions and few or many decision variables. The following paragraphs outline a few example problems indicating the wide range of applicability of the tuning process. The tuning process was not applied to these problems in this thesis but they are examples in the literature of uses of parts or all of the process. A detailed example of the tuning process for a finite element model will be given in the next chapter.

A basic example showing the use of the optimization technique

(SUMT) used in this thesis is found in Ref 5. The example has one objective function, two inequality constraints and three variables. With only one objective function, only one pass through SUMT is required to determine the optimum values of all three variables within the constraints. In PROCES, this would not require the multiple objective software but would give a NDSS with one member, assuming the global optimum was found.

A simple example of the use of PROCES on a vector problem is seen in Ref 2. The interactive version of PROCES used contained an optimization technique for linear problems only with the objective functions in the form of single mathematical statements. The vector optimization problem used three objective functions and five decision variables. The two objective functions used as constraints were varied between their lower and upper bounds such that 121 combinations of the two were possible. After the 121 passes through the optimization routine to form the 121 candidate solutions, a NDSS with 81 members was found. The 81 member NDSS was then rank ordered by two different weighting combinations but without the normalization implemented for this thesis. Each weighting combination produced a most preferred solution.

Students in the Graduate Systems Engineering program at the Air Force Institute of Technology have been using PROCES to optimize missile and spacecraft designs. Ref 6 examines the case of a missile design using a PROCES derivative called Advanced Effectiveness System Optimization Program. Ref 7 is the use of PROCES to design a spacecraft. For these two problems, diverse performances such as cost, survivability and capability have been used as objective functions with

some performances being maximized and some minimized. Data for these problems were simulated by single statement objective functions.

An example finite element model problem is outlined below in which six objective functions and three decision variables were used. The objective functions were in the form of tabulated data. The resulting NDSS contained over 200 members which could be lumped into five unique solutions depending on the relative importance of each objective function.

III Finite Element Model Example

Introduction

This example of the tuning process will show how a finite element model can be adjusted to reproduce experimental results. This problem is one of current interest to the Air Force and represents how a real problem can be analyzed with the procedure outlined in this thesis. The presentation includes the model tuning results and the analysis of the NDSS.

History

The Purpose. The United States Air Force needs an improved flutter prediction capability for horizontal stabilators. The T-38 Talon jet trainer stabilator is such a structure. San Antonio Air Logistics Center (SAALC) carries the responsibility for the engineering and maintenance of these aircraft, repairing any structural damage on the stabilators. The stabilators experience damage such as delamination of the skin, holes, cracks and corrosion. Mass is added to the stabilator during the repair since the repair material used in place of the removed portions of core is more dense. This additional mass, as well as the damage itself, will typically decrease the flutter speed. Because of flutter degradation limits, SAALC has had to replace many stabilators rather than reuse them, running the spare stabilator inventory quite low and costing the USAF thousands of dollars. SAALC's current analysis method is somewhat conservative, resulting in only small repairs being allowed. Possibly, by using a more accurate degradation prediction technique, more stabilators can be repaired and

reused.

SAALC began looking for a more accurate flutter speed prediction capability in 1975. AFIT became involved in this project in 1979. Several AFIT Aeronautical Engineering Master of Science degree students have been examining various aspects of the project. This example will further extend the effort by showing how the model can be tuned to better simulate the stabilator. The tuned model will be a key piece in the overall effort to assemble a more accurate flutter analysis capability for the T-38 stabilator.

SAALC started their effort to improve their flutter speed degradation prediction capability over their current method. The method in use now calculates the unsteady aerodynamic forces on the stabilator using Strip Theory (Ref 8). The wing is divided into numerous chordwise sections, or strips, across the span. The aerodynamic loads are determined based on the two-dimensional aerodynamic coefficients down the middle of the strip. The strips do not interact but can rotate (pitch) and move laterally (plunge). Thus the motion of the wing is determined using bending and torsional modes. Chordwise deformation of a section is not considered. Newer analysis methods, such as the Doublet-Lattice Method, incorporate chordwise deformation also. Results of flutter analysis using Strip Theory generally yield conservative flutter speed decrements, leading to much lower absolute flutter speeds than actually exist. Utilizing the newer technology flutter speed analysis techniques incorporating the Doublet-Lattice Method in a finite element computer program may improve the flutter speed predictions.

AFIT thesis students have completed four investigations into an

improved flutter speed prediction technique for SAALC. The two versions of the T-38 stabilator were the subjects of these investigations. The emphasis was placed on the Series 3 stabilator since this version is currently in use (see Appendix A). John O. Lassiter (Ref 9) began the projects by setting up the analysis procedure using a finite element program and beginning work on verifying the model. Roger K. Thomson (Ref 10) then compared the model's static deflections to the Series 2 stabilator documented deflections. He also measured the Series 3 stabilator's natural frequencies and then predicted the frequencies using the finite element model. These investigations led to the conclusions that the model needed some adjustments to properly simulate the stabilator. Lex C. Dodge (Ref 11) and Jack O. Sawdy (Ref 12) continued the investigations. Dodge manually tuned the model to match the Series 3 stabilator's mode shapes and frequencies while Sawdy measured the Series 3 stabilator's static displacements and tuned the model to these. No one, though, tuned the models for both the static and dynamic data at the same time.

Lassiter used the finite element technique embodied in NASTRAN (NASA Structural Analysis computer program) as his basic analysis tool (Ref 13). NASTRAN was used to analyze the static deformations of the stabilators under load conditions and determine the structures' natural frequencies and mode shapes. Lassiter first developed a two-dimensional finite element model of the stabilator, as described in Appendix A, to represent the mass and stiffness distributions. He found the torque tube to be a particularly difficult item to model due to its varied characteristics. Using NASTRAN, he determined the

model's static displacements for given load conditions and found that these were close, but did not match those of the Series 2 stabilator. Lassiter then obtained the natural frequencies and mode shapes of the model with NASTRAN. These, also, were close to the desired values. The flutter analysis procedure requires an aerodynamic model and an unsteady aerodynamic analysis method, such as Strip Theory, Doublet-Lattice Method, Mach Box Method, or Piston Theory, to be chosen. For these efforts, the Doublet-Lattice Method was used to produce the aerodynamic coefficients for the unsteady flow. The flutter speed predictions, however, did not agree with other sources. It was believed that the model needed to be adjusted to reproduce the measured data. However, no static displacement data existed for the Series 3 stabilator for comparisons with the model's data. Also, the aerodynamic model had not been verified.

Roger K. Thomson examined both the finite element model and the aerodynamic model. By increasing the elastic and shear moduli of the model's elements, he was able to improve the model's reproduction of the static displacements for the Series 2 stabilator. This analysis could not be performed for the Series 3 stabilator because no static deflection data existed for that stabilator. Thomson analyzed the aerodynamic model by comparing it to steady windtunnel data. He found the model agreed very well with the limited experimental data. Finally, he performed a modal analysis on the Series 3 stabilator both experimentally and analytically. He measured the free-free modes and natural frequencies of the stabilator and compared them with NASTRAN generated data. The NASTRAN frequencies were lower than the experimentally determined frequencies. However, the mode shapes agreed

well. It was believed that an improved model could possibly remove the discrepancies in the frequencies and static displacements.

Dodge's goal was to tune the finite element model so that it would reproduce the experimental frequencies and mode shapes. Since the model's frequencies are used in the flutter analysis, a more accurate frequency model would hopefully lead to a better flutter speed prediction. Thomson's experimental frequency values and mode shapes were the reference values for Dodge's effort. Dodge manually varied element parameters to force the model to reproduce the experimental frequencies and mode shapes. The parameters varied included the material properties of modulus of elasticity and shear modulus for the plate and bar elements; the cross-sectional properties of bending and polar moments of inertia for the spar, leading and trailing edge elements; the mass density of all the elements; and the plate thickness. The parameters were varied based upon engineering judgement, which improved as more insight was gained in how the parameters effected the model's frequencies and mode shapes. Dodge discovered that increasing the plate thickness by 37% while holding all the other parameters at their original values produced the model which best simulated the stabilator in the free-free vibration condition. This model also did very well with the aircraft installed boundary conditions when compared to the measured frequencies.

Dodge also varied the stabilator pitch spring value for the two different boundary conditions. Two sets of boundary conditions were created by the different installation attachments between the aircraft installed condition and the test configuration. When the stabilator is installed in the aircraft, the pitch spring stiffness is a combination

of the aircraft's hydraulic control system, actuator arm and torque tube stiffnesses. This control stiffness varies depending on whether one or two aircraft hydraulic systems is in operation. For the test conditions, the pitch spring stiffness had contributions from the actuator arm attachment (a reaction load measuring device), the actuator arm and the torque tube. The best control stiffness for the aircraft conditions was 4.4×10^6 in-lbs/rad (see Appendices B and C), which was chosen to correct the first torsional frequency. See Table I below for a summary of the results. The first four natural frequencies are listed as the first bending mode (lowest natural frequency mode), first torsional mode, second bending mode and second torsional mode. The free-free boundary condition was measured by Thomson (Ref 10). The aircraft installed condition frequencies were measured and calculated by NAI (Ref 14) and are also in Table I. The Dodge model data refers to Dodge's finite element model as hand tuned to the free-free condition's first torsional frequency. The model tuning accomplished by Dodge, though, did not consider the static displacements and did not allow for the model tuning to be done in a mechanized procedure.

TABLE I
Measured and Calculated Frequencies

<u>Mode</u>	<u>Free-Free Condition</u>		<u>Aircraft Conditions</u>		
	Thomson Measured Hz	Dodge Model Hz	NAI Measured Hz	NAI Calculated Hz	Dodge Model Hz
1st Bending	- -	- -	17.3	17.61	17.45
1st Torsion	100	105.5	44.9	44.89	44.41
2nd Bending	124	121.0	71.7	78.76	75.11
2nd Torsion	160	149.8	- -	- -	- -

Sawdy investigated the model's static performance. He experimentally determined the static displacements of the Series 3 stabilator under 10 different load conditions. He measured the stabilator displacement at 25 surface positions for each of the 10 load conditions and then tuned the model to these displacements. Sawdy, like Dodge, used engineering judgement to determine how to change the parameters. He introduced orthotropic plates to better model the differences between chordwise and spanwise deflections. Sawdy then varied the skin's chordwise and spanwise Young's Moduli, the torque tube area moment of inertia and the plate thickness. He found good results by increasing the chordwise modulus by 200% and decreasing the inertia of the outboard section of the torque tube by 25%. He formulated a mechanized method to perform this tuning process using the multiple objective optimization technique (MOOT) to allow choice of the best model in the presence of conflicting goals. This formed the basis for the present work.

Objective Functions

The Performance Errors. The performance indices for this effort consist of the displacement and natural frequency errors between the predicted and actual values. Displacement errors between the model and real stabilator for identical boundary and loading conditions and at the same measurement locations tell how well the model simulates the structure's static load performance. Likewise, the differences between the natural frequencies calculated and measured are figures of merit for a dynamic model. Sawdy and Dodge developed good static and dynamic models, respectively, and these will be the starting points for the tuning in this example.

Displacement errors for four different conditions were derived from available data. Sawdy measured the displacements of the stabilator at 25 different locations for each of 10 different load conditions. Adding the error from all 10 loading conditions together gives another set for a total of 11 sets of measured displacement data. NASTRAN can predict corresponding displacement values for any particular model, then compute the difference between the two sources of data. Four of the 11 were selected for use in this thesis in order to keep the problem small for ease of data handling and comprehension. Also, since seven of the 11 loading conditions were combinations of the other four, only four were necessary to represent the entire set. Sawdy's cases 2, 4 and 10 and the combination of all 10 cases were the four cases selected. Case 2 was an example of spanwise bending, case 10 was chordwise bending with torsion and case 4 a mixture of the two. The displacement performance functions were formed from the square root of the sum of squares of the 25 displacement errors for each case. Mathematically, that is:

$$Z_j(\underline{X}) = \left\{ \sum_{i=1}^{25} \left[w(\underline{X})_{ij, \text{NASTRAN}} - w(\underline{X})_{ij, \text{EXP}} \right]^2 \right\}^{\frac{1}{2}} ; j=1,2,\dots,10$$

where $w(\underline{X})_{ij, \text{NASTRAN}}$ is the displacement for the j th loading condition as predicted by NASTRAN and $w(\underline{X})_{ij, \text{EXP}}$ is the corresponding experimentally determined displacement. Also, the addition of all 10 displacement errors gives:

$$Z_{11}(\underline{X}) = \sum_{j=1}^{10} Z_j(\underline{X}).$$

These performance indices indicate how accurately a model predicts the static displacements of the structure. An error of zero would be obtained for the perfect model.

Two frequency indices were similarly obtained for use in the tuning procedure. The first three natural frequencies for the stabilator were measured and published by the aircraft manufacturer (Ref 14). The differences between these three values and the predicted values form three more candidate performance indices. Summing all three gives a fourth. We have:

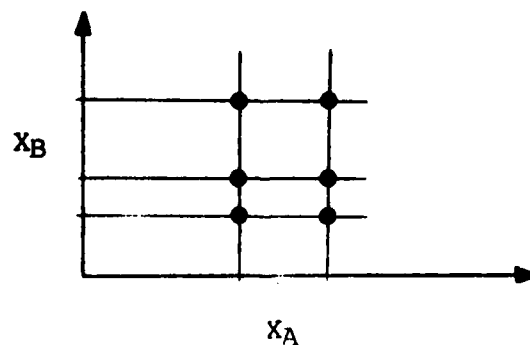
$$\omega_j(\underline{X}) = \left| \omega(\underline{X})_{j, \text{NASTRAN}} - \omega(\underline{X})_{j, \text{EXP}} \right| ; \quad j=1, 2 \text{ or } 3$$

$$\text{and } \omega_4(\underline{X}) = \left\{ \sum_{j=1}^3 \left[\omega(\underline{X})_{j, \text{NASTRAN}} - \omega(\underline{X})_{j, \text{EXP}} \right]^2 \right\}^{\frac{1}{2}} .$$

The first three natural frequencies are the first bending, first torsion and second bending modes. The first torsional mode was thought to be a better indicator of model performance since this mode most influences flutter. Therefore, the first torsion and the total frequency error were the two frequency indices chosen for this problem. Again, the better the model, the lower the performance error. Due to the complexity of the finite element model, though, the best frequency model may not also be the best static model. In fact, as will be seen later, these two models are at opposite corners of the decision variable space.

Mechanization via PROCES. The performance indices were put into PROCES in the form of data tables. Appendix C illustrates how this was accomplished. The limits of the tables represent the boundaries of the

decision variable space. The decision variable space was established such that it contained the original values of the variables and Sawdy's static model. Dodge's frequency model could not be entirely contained because of its increased thickness ratio. The decision variable space was represented by a grid network with each grid point corresponding to a different model. This grid network filled with models is illustrated below for two decision variables. The decision variables X_A and X_B each range from a minimum value to a maximum value. Within those ranges, two values were chosen for X_A and three for X_B . The variable space is represented by the six intersection points of the two variables. The coordinates of each point are a specific combination of decision variables representing a particular model. Using NASTRAN and the measured data with the variable space for the finite element model, the performance errors for each intersection point were found and placed in the tables. Each performance index had its own table with the dimensions equalling the dimensions of the corresponding variable space.



A table look-up routine was employed to determine the performance errors corresponding to a particular set of decision variables. The table look-up routine could determine the performance errors for any

model within that space, not just the grid points, so that the performance errors would be continuous. A linearly interpolating table look-up routine was used with a small table. A higher order routine or a more dense table would have improved the accuracy of the interpolated performance errors as obtained from the precalculated values. Sparse tables were chosen to reduce the time required to prepare the data for them. Examining the members of the NDSS with the weighted performance index method or with decision variable graphs will indicate the regions which contain the appropriate solution. These regions can then be expanded in the tables or individual models can be rerun with NASTRAN to get more accurate performance errors, if needed.

Although the decision variable space is continuous, the solution space may not be. The decision variable space is complete with no holes or gaps and will produce a performance set for any point within the decision variable space. However, this does not guarantee the continuity of the entire performance space. The performance space could have voids or holes in it because of the nonlinear dependence of the errors upon the parameters. In other words, there could be performance error combinations which are infeasible and have no corresponding X .

Decision Variables

Three material and physical properties were chosen as the decision variables. The variables chosen were the area moment of inertia for the torque tube and the spanwise and chordwise Young's Moduli for the orthotropic plate elements. These were global variables with the same percentage change in moment of inertia applied to all three portions of the torque tube and the Young's Moduli changes applied equally to all

plate elements. The small number of decision variables was chosen to keep the problem manageable. These particular variables were selected since they were the most influential variables examined in the other four investigations for fine tuning the model. The spring constant and the thickness to chord ratio were manually tuned by other students to produce good results. The variables were given the freedom to change above and below a reference value. The reference value was established by the physical dimensions or the actual material properties of the structure. Although all of these decision variables will have some effect on both the static displacement errors and the frequency errors, it should be expected that the effects will be conflicting. This is evidenced by the two different models already developed. Any solution to the problem, then, should be a compromise solution somewhere in between the best static and best frequency models in the three-dimensional decision variable space.

Constraints

The inequality constraints of the problem are the limits of the decision variables. The three decision variables chosen are all required to be positive for the model to be realistic. Although this sets a lower limit for each variable, it will be trivially satisfied due to the next, more stringent set of limits. This problem has the practical limits of the performance error table boundaries. The large number of manhours required to submit numerous runs to prepare the performance error tables and then to analyze and make use of all this data sets practical limits on the table sizes. Extrapolation outside the tables is possible but is subject to much uncertainty. Therefore, the objective functions were required to remain on the table interiors,

making the table boundaries the decision variable limits.

The constrained objective functions produce equality constraints. These objective functions are constrained to different prescribed values each pass through the optimization procedure. The lower and upper bounds specified by the analyst should be the true problem performance error limits as evidenced in the performance error tables. These two types of constraints, then, limit the vector space which can solve the problem and the performance space which indicates where the solution lies.

Finite Element Model Results

Tuning the three decision variables to minimize the six static and frequency errors produced five unique models. The five models were determined from the NDSS by systematically cycling through many combinations of the relative weightings between the static and frequency errors. One model was predominantly a static model, one a frequency model and the other three were compromises of the two.

The performance values were obtained by exercising NASTRAN on each model (or grid point) in the decision variable space. This three-dimensional space for the static cases was described by four values of the torque tube area moment of inertia, three values of the plate chordwise Young's Modulus and four values of the plate spanwise Young's Modulus. The 4 by 3 by 4 tables, then, contained 48 models to be examined by NASTRAN. The frequency data was handled using three values of the torque tube inertia, three values of the chordwise Modulus and five values of the spanwise Modulus. These 3 by 3 by 5 tables created 45 NASTRAN runs. Therefore, the four static deflection error tables and the two frequency error tables were obtained from 93

NASTRAN runs. The performance tables are shown in Appendix B.

Starting points from opposite ends of the decision variable space were used with the first developing 202 members in the NDSS and the second 255 members. All the members of these two large sets were variations of a few unique models as the performance values were stepped methodically between their upper and lower bounds. Figures 6 and 8 in Appendix D show where the NDSS members are located within the decision variable space. Figure 7 is a two-dimensional view of Figure 6. The coordinates of each \underline{x} represent the values of the decision variables (defining a model) of a non-dominated solution. These three figures show graphically how the optimal solutions reflect in the variable space. One can see the gaps in the space in which solutions do not exist and how the solutions bunch into certain feasible regions and track along the edge of a hole. Various combinations of weights were then applied to these NDSS's and five unique models were found. As the weights were varied within a certain range, the two highest ranked models were always in a particular group of solutions. The approximate midpoint of that group was then used as the answer for that weighting range. Five ranges and solutions were determined in this manner. These five groups of models are circled in the figures. Appendix D's Table IX lists the five final models with their corresponding weighting ranges.

Each of the five final models is associated with a different relative weighting between the static and the frequency cases. To form a comparison with the combined problem's best static and best frequency models, the static cases were tuned without the frequency cases and the frequency cases without the static cases. The tuned static case only

model was almost the same as the combined problem's best static model which had 85 to 100% of the weight applied to the four static cases. Without any influence from the frequency considerations, the static model had a chordwise Modulus of 25% over the reference value compared to 10% for the combined problem's static model. The frequency only models were identical. The other three models were achieved by observing the best solutions for many different combinations of weightings in between the best static and best frequency models. As seen in Table IX, 60 to 80% of the weight on the static cases produces a unique solution, 25 to 55% another solution and 20% the third. The NDSS contained all of the possible answers, although not all members of the NDSS fell into one of the five best answers for this weighting system. Now the decision maker could select which of the models is best for his situation by applying his own preferred weighting criteria and obtaining one of the five models as his solution.

Several conclusions can be deduced from the five models. First, the five models are dependent only on the difference in ranking between the static and frequency cases. The rankings between the four static cases or the two frequency cases were insignificant. The model could have been tuned using only two objective functions instead of six, one static load deflection condition and one natural frequency condition. Either an individual load or frequency case error could have been used or a total load or frequency error. In addition, the spanwise modulus was higher as more weight was applied to the frequency performance. However, the chordwise modulus was unaffected by the weights except for a 65% change for the pure static model. Also, the decision variable limits used were too restrictive. Four of the five models had one or

more decision variables on or very near a limit. It is desirable to have all models within the feasible region in order to be absolutely sure the global minimum is chosen. The table limits should have been wider. Finally, although two very different starting points were used in the optimization process, the same models were obtained. For this problem, it appears that the global minimum within the table boundaries was found.

07 The five unique models obtained as the stabilator solution can also be compared to the models hand tuned by Dodge and Sawdy. Table IX also lists the pertinent data for each model. The PROCES tuned static model is very similar to Sawdy's model in total deflection error. The two models, however are not very similar in physical properties, as can be seen by comparing the decision variables. This substantiates the statement made earlier that more than one X could produce the same set of performance indices. Although these two models seem to be equally good, the PROCES tuned model had the added influence of the frequency constraints during the tuning procedure whereas the Sawdy hand tuned model did not. Thus, the PROCES tuned static model may be a better model for predicting static and frequency data than the hand tuned Sawdy model.

The PROCES frequency model and the Dodge model are very different. Dodge varied the thickness to chord ratio instead of the three decision variables used in the PROCES tuning procedure. The resultant physical properties are much different than PROCES's. The PROCES tuned model had the better total frequency error by 66% but a larger first torsional mode error by 97%. Here, as in the static case, the hand tuned model did not consider the static deflections during the tuning

procedure whereas the PROCES tuned model did. If minimizing the first torsional mode error is more indicative of the correct flutter speed, then the Dodge model should be selected as the solution. Otherwise the PROCES tuned model would be the better choice. Some other criteria such as a flutter analysis may also be required to determine which of the two models is preferred.

A flutter analysis was performed on three of the PROCES tuned models. London (Ref 15) is improving the NASTRAN flutter analysis procedure for repaired T-38 stabilators in a concurrent effort. He has compiled a list of target flutter data for the stabilator from available sources as shown in Appendix D in Table X. His analysis of Dodge's model shows that the model matches the target data very well. London also examined the PROCES tuned best static model, best frequency model and best first torsional mode model. The best first torsional mode model was that model which had the lowest first torsional mode error without any influence from the other frequency errors or the static deflection errors. This model was not tuned by the MOOT procedure. The flutter analysis results of these models are shown in Table X.

The PROCES tuned models did not match the estimated flutter data very well. The two best frequency models produced the correct flutter frequencies but not the proper flutter speeds. Conversely, the best static model had a more accurate flutter speed but worse flutter frequency. The untuned first torsional mode model matched the target data just as well as the tuned frequency model. Perhaps a model tuned to a static error case and the first torsional mode would produce a better solution. Also, other decision variables such as the thickness

to chord ratio used in Dodge's model may be necessary in the PROCES tuned model to reduce the errors.

The automated tuning process carried out above allowed a much more thorough analysis to be completed than could have been accomplished manually. The optimization routine examined hundreds of models in building the NDSS which contained over 200 members and completed the entire task very quickly. Manual optimization requires the analyst to have keen insight into the problem which he may not have or have time to acquire. The automated process does not require as much knowledge of the problem.

The analyst assisted weighting scheme provided a good model selection method allowing for the possibility of diverse decision maker opinions and rapidly changing situations. This adds the flexibility of quickly selecting the best answer without having to reaccomplish the entire tuning procedure.

The sparse performance error tables were appropriate for this tuning procedure. As can be seen in Appendix D in Table IX and Figures 6, 7 and 8, the weighting system located only a few (five in this example) models within the decision variable space which were desirable solutions to the problem. Since many models exist in the same region but are represented as one desirable solution, the individual model accuracies are lost in the averaging procedure. Therefore, interpolating within a denser (more accurate) table and using a higher order interpolation scheme to give more accurate models, are not warranted.

Extensions of Example

Modal Tuning. Mode shape errors may be a better performance index

than the natural frequencies for tuning the model for flutter speed prediction. The natural frequency was chosen for this thesis to simplify the tuning process. The mode shapes were never examined. Dodge (Ref 11) manually tuned the frequency model to the natural frequencies and corresponding mode shapes. He used visual inspection to determine the mode shape accuracy. Since the mode shape is based on a physical deflection envelope, these deflection errors could become the objective function or additional objective functions. These objective functions could be implemented like the static deflection errors were.

An alternate method of making use of the mode shapes would be to incorporate the mode shape as a constraint and maintain the natural frequency as a performance index, or vice versa. This type of constraint would require another table look-up but could be accomplished just like the objective functions were for the example problem. Now the mode shape error (or natural frequency error) would be kept small while the natural frequency error (or mode shape error) became one of the performance indices to be minimized.

Repair Tuning. The repair could be tuned to possibly reduce the flutter speed decrement once the stabilator model is chosen. In this case, the finite element model problem decision variables would be replaced with the repair material properties and physical dimensions. The performance table would be flutter speed degradations for various combinations of the decision variables. Again, NASTRAN could be used with the appropriate model properties to compute the flutter speeds to fill in the table. Rather than using the smallest repair possible, perhaps changing the repair's shape or material properties or

additionally changing the stabilator's properties at some other location would reduce the flutter speed degradation. Here, though, limitations may have to be imposed due to repair costs, machinability or stabilator static strength which will constrain the size or shape of the repair that is possible. The optimization process could be carried out just like the tuning of the finite element model except now there would be only one objective function, the flutter speed degradations. In this manner, the repair which will give the minimum flutter speed decrement can be found.

IV Conclusions and Recommendations

Conclusions

A method to tune finite element models to reproduce selected experimental data was assembled. The method is automated, not requiring the user to guess at how to improve the model. The process incorporates multiple objective optimization theory and a weighted performance ranking system. The process performs smoothly, although much time and effort has to be expended to construct the data tables. The method is versatile and can be easily adapted to various types of model tuning or optimization problems.

A T-38 stabilator was tuned as an example. Of the over 200 members of the two non-dominated solution sets, only five models were unique solutions to weighted combinations of static and frequency performances. In this example, there was no single solution which simultaneously performed the static load conditions and the frequency conditions best. The best static model, best frequency model and the three intermediate combinations of these two are listed in Appendix D in Table IX.

Recommendations

All the recommendations involve or are derived from the T-38 stabilator example. In this example, only the static deformation load case errors and the first three natural frequency errors were used as figures of merit. The mode shape may tune the model for flutter predictions better than the static deformations or frequencies did. The mode shape, then, should be investigated as the performance index.

Also, the table limits should be extended to keep all the solutions on the inside of the decision variable space. This insures that the solution doesn't lie outside the space whenever an answer is found near or on a boundary. This recommendation should be followed for all problems. Finally, for this particular stabilator example, other variables should be used which would reduce the performance errors to zero. None of the decision variable combinations used in this example appreciably zeroed any of the static errors. Perhaps the thickness to chord ratio should be added as a decision variable.

Bibliography

1. DeWispelare, A. R. "Algorithm Efficiency in Generating Non-Dominated Solution Sets," Proceedings of the IEEE 12th Annual Symposium on System Theory. Virginia Beach, Virginia: 1980. Pages 218-222.
2. DeWispelare, A. R. Multiple Objective Optimization Program Extensions - PROCES. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, unpublished.
3. Mylander, W. C., Holmes, R. L., and McCormick, G. P. A Guide to SUMT-Version 4: The Computer Program Implementing the Sequential Unconstrained Minimization Technique for Non-linear Programming. Research Analysis Corporation, McLean, Virginia, 1971.(AD 731 391)
4. Wismer, D. A. and Chattergy, R. Introduction to Nonlinear Optimization. New York: Elsevier North Holland, Inc., 1978.
5. Fiacco, A. V. and McCormick, G. P. Nonlinear Sequential Unconstrained Minimization Techniques. New York: John Wiley and Sons, Inc., 1968.
6. Gibson, R. B., et al. Investigation of the Feasibility of a Conventionally Armed Tactical Cruise Missile. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1981. (AD C027 173/4)
7. Robinson, D.G., et al. Conceptual Feasibility Study of an Advanced Military Spacecraft Capability. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1981. (AD C027 170/1/2)
8. Scanlan, R. H. and Rosenbaum, R. Introduction to the Study of Aircraft Vibration and Flutter. McMillon, 1951.
9. Lassiter, John O. Initial Development for a Flutter Analysis of Damaged T38 Horizontal Stabilizers Using NASTRAN. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, March 1980.
10. Thomson, Roger K. Investigation of an Improved Flutter Speed Prediction Technique for Damaged T-38 Horizontal Stabilizers Using NASTRAN. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1980.
11. Dodge, Lex C. Investigation of an Improved Structural Model for Damaged T-38 Horizontal Stabilizer Flutter Analysis Using NASTRAN. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1981.

12. Sawdy, Jack O. Investigation of a Two-Dimensional Model for the Prediction of Static Displacement for T-38 Horizontal Stabilators Using NASTRAN. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1981.
13. NASA SP-222(05). The NASTRAN User's Manual, Level 17.5. Washington DC: Scientific and Technical Information Division, National Aeronautics and Space Administration, December 1980.
14. NAI-58-6. T-38A Vibration Mode Analysis of Empenage. Hawthorne, California: Northrup Aircraft, Inc., May 1959.
15. London, George G., Jr. Investigation of an Improved Finite Element Model for a Repaired T-38 Horizontal Stabilizer Flutter Analysis Using NASTRAN. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1982.

Appendix A

The T-38 Stabilator

The T-38 Series 3 stabilator built by Northrop Aircraft, Inc. is the subject of the example presented in Section III. Figure 1 is a picture of the stabilator. The stabilator has a circular cross-section torque-tube with one main spar, a root rib, a tip rib and a leading edge plate. The honeycomb core is sandwiched by aluminum skin. The Series 2 stabilator was an earlier version with three additional ribs and another spar.

The Series 3 stabilator was modelled by a two-dimensional finite element model. The spar, ribs, and torque-tube were represented in NASTRAN by bar elements and the skin/honeycomb core combination by either quadrilateral or triangular plate elements. See Figure 2 for an exploded view of the model. The model's elements were adjusted with triangular sections and nodes out of uniform position to locate nodes where loads were measured during the experimental tests. A computer program generated most of the data describing the model for input to NASTRAN. References 9, 10, 11 and 12 describe the model more fully. The same model was used for both the static and the natural frequency cases. The model contained 166 grid points, 95 quadrilateral membrane and bending elements, 80 triangular membrane and bending elements and 74 simple beam (bar) elements. Of the 898 possible degrees of freedom, five were restricted to match the boundary conditions.

The static and frequency cases required different boundary

conditions. The different conditions are caused by the different methods in which the structure was attached to the test apparatus for the static deformation measurements and the aircraft installed conditions for the frequency measurements. The test conditions for the static case create an asymmetric loading condition with the boundary conditions specified at the actuator arm and bearings. The spring constant for the load cell at the actuator arm was held constant at the best value of 1.18×10^6 in-lb/rad as determined by Sawdy. Rotations in all directions at both bearing nodes were allowed but only movement in the torque-tube axis direction at the outboard bearing was permitted. For the frequency case, the spring constant for the hydraulic actuation system was fixed at 4.40×10^6 in-lb/rad as found by Dodge. Rotations and torque-tube axis movements were allowed at the outer bearing. The torque-tube was permitted free movement on the vehicle plane of symmetry. The torque-tube was also allowed to twist about its own axis at the centerline since the pitching moment was taken out through the pitch spring and actuator arm.

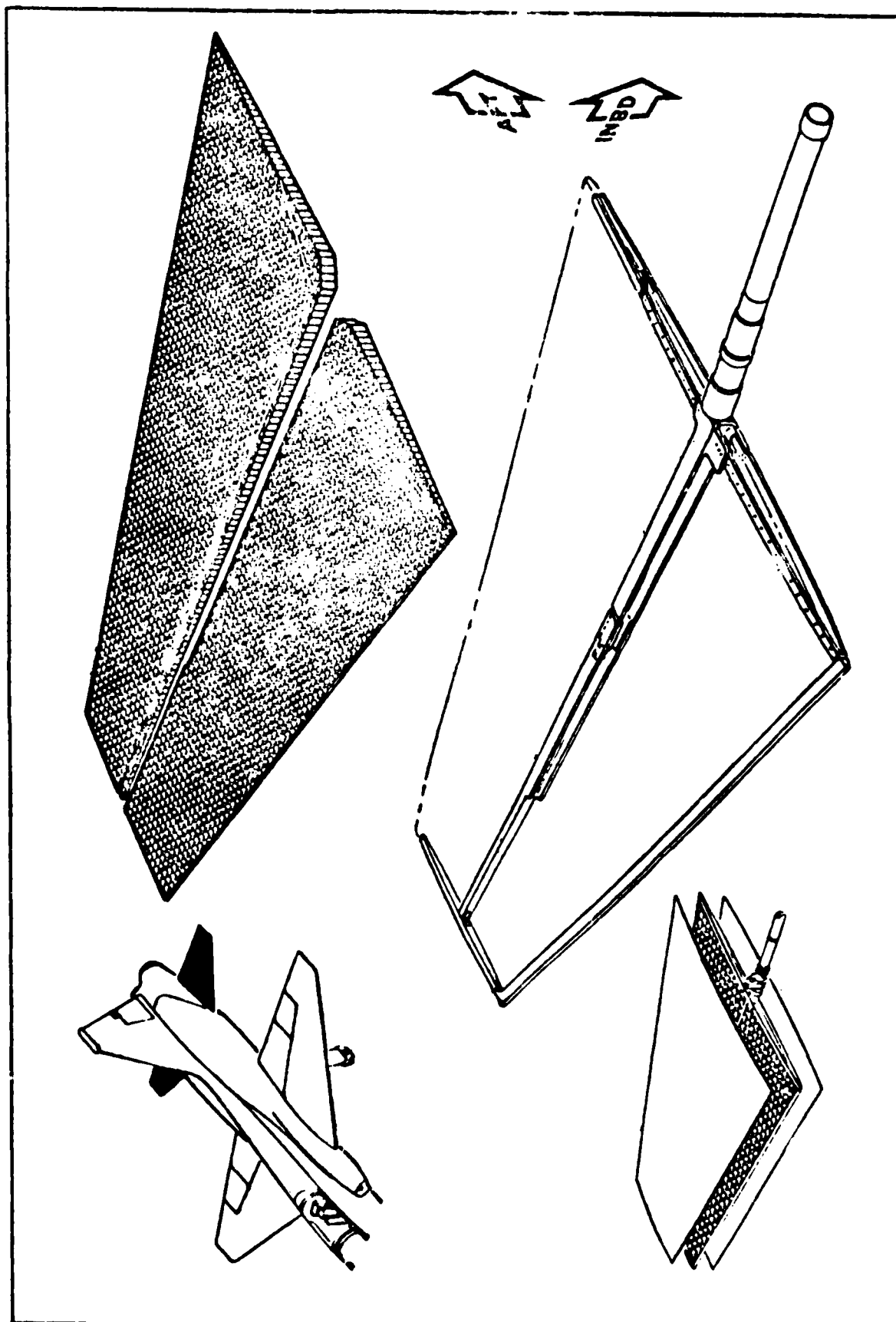


Figure 1. T-38 Series 3 Stabilator

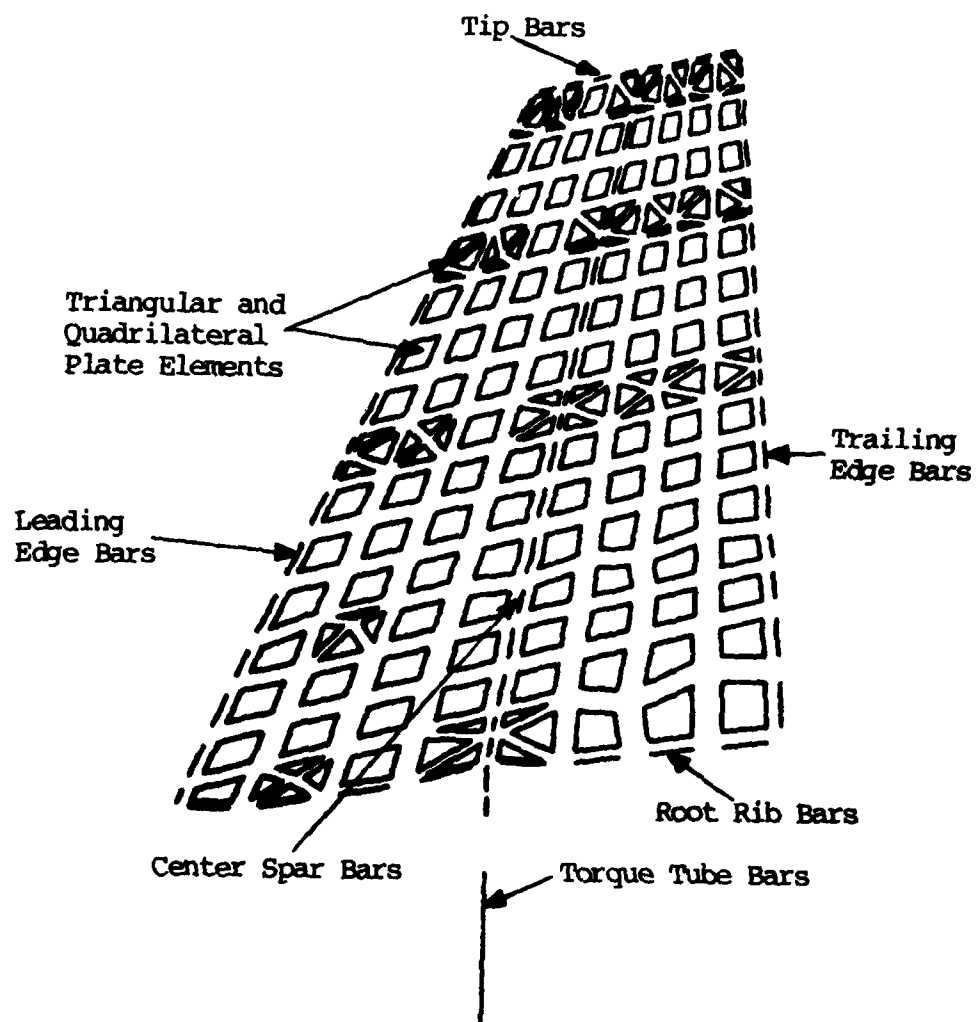


Figure 2. Exploded Finite Element Model of the T-38 Horizontal Stabilator

Appendix B

Displacement and Frequency Data for the T-38 Stabilator

07 The performance errors for each grid node in the decision variable space were calculated with NASTRAN. The decision variables were systematically varied over the space with each set being the input parameters for the data generation program which produced the NASTRAN input data file. Table II lists the decision variable values used for both the static load cases and the natural frequency cases. Forty-eight possible combinations were used for the static analysis and 45 for the frequency cases. NASTRAN computed the load case errors and total load case error for each combination of variables for the static analysis and the first three natural frequencies for each combination of variables for the dynamic analysis. The static load case deflections found by NASTRAN were compared to those measured by Sawdy using a DMAP ALTER routine in NASTRAN as discussed in Ref 12. The natural frequencies estimated by NASTRAN were compared to the NAI measured frequencies to determine the errors. The performance errors for the four static cases and two frequency cases are shown in Tables III through VIII. The unit of the static errors is inches of displacement and cycles per second for the frequency errors.

TABLE II

Static and Frequency Decision Variable Values*

	Torque Tube I _y	Chordwise Modulus	Spanwise Modulus
Static Cases	.75	.75	.75
	.80	1.25	1.25
	.875	1.75	1.75
	1.0		2.75
Natural Frequency Cases	.75	.75	.75
	.875	1.25	1.25
	1.0	1.75	1.75
			2.25
			2.75

* The factors listed are multiplicative factors of the reference values of 3.04345, 1.8797 and 3.72 in⁴ for the area moments of inertia terms for the three torque-tube sections and 11.56×10^6 psi for the chordwise and spanwise Young's Moduli.

TABLE III

Total Static Load Case Errors (In.)

Torque Tube Iy	Chordwise Modulus	Spanwise Modulus			
		0.75	1.25	1.75	2.75
.75	.75	4.2171	1.9802	2.3890	3.6716
.75	1.25	3.6294	1.7836	2.4857	3.9020
.75	1.75	3.3011	1.8317	2.6850	4.1328
.80	.75	3.6652	1.8330	2.6214	4.0701
.80	1.25	3.0947	1.7813	2.8093	4.3533
.80	1.75	2.7994	1.9356	3.0514	4.6001
.875	.75	3.0190	1.9429	3.0832	4.6829
.875	1.25	2.5049	2.0793	3.3656	5.0132
.875	1.75	2.2746	2.3211	3.6333	5.2690
1.0	.75	2.3661	2.4955	3.8931	5.6182
1.0	1.25	2.0023	2.7961	4.2463	5.9792
1.0	1.75	1.8975	3.0829	4.5279	6.2396

TABLE IV

Static Load Case 2 Errors (In.)

Torque Tube Iy	Chordwise Modulus	Spanwise Modulus			
		0.75	1.25	1.75	2.75
.75	.75	.38278	.12708	.22772	.41376
.75	1.25	.35279	.12722	.24128	.42485
.75	1.75	.32947	.13000	.25270	.43402
.80	.75	.33695	.12532	.26218	.45472
.80	1.25	.30712	.13322	.27689	.46603
.80	1.75	.28398	.14177	.28914	.47539
.875	.75	.27863	.14711	.31137	.50892
.875	1.25	.24915	.16206	.32694	.52045
.875	1.75	.22645	.17526	.33983	.52997
1.0	.75	.20287	.20235	.38204	.58313
1.0	1.25	.17456	.22137	.39822	.59483
1.0	1.75	.15334	.23720	.41157	.60449

TABLE V

Static Load Case 4 Errors (In.)

Torque Tube Iy	Chordwise Modulus	Spanwise Modulus			
		0.75	1.25	1.75	2.75
.75	.75	.41708	.12581	.16172	.34084
.75	1.25	.33485	.09029	.20816	.39719
.75	1.75	.28742	.10955	.24924	.43674
.80	.75	.37226	.10544	.19525	.38296
.80	1.25	.29064	.10183	.24859	.44079
.80	1.75	.24465	.13632	.29094	.48064
.875	.75	.31559	.10757	.24499	.43860
.875	1.25	.23543	.14046	.30340	.49778
.875	1.75	.19264	.18319	.34658	.53787
1.0	.75	.24296	.15353	.31719	.51451
1.0	1.25	.16764	.20882	.37919	.57489
1.0	1.75	.13482	.25479	.42286	.61515

TABLE VI

Static Load Case 10 Errors (In.)

Torque Tube Iy	Chordwise Modulus	Spanwise Modulus			
		0.75	1.25	1.75	2.75
.75	.75	.24745	.19775	.16603	.13030
.75	1.25	.14190	.11127	.10102	.10533
.75	1.75	.11762	.10932	.11602	.13692
.80	.75	.21243	.16809	.14208	.11872
.80	1.25	.11295	.09927	.10384	.12490
.80	1.75	.10639	.11869	.13607	.16569
.875	.75	.18220	.15189	.13953	.13841
.875	1.25	.10973	.12399	.14247	.17368
.875	1.75	.12983	.16001	.18381	.21776
1.0	.75	.18065	.17751	.18401	.20311
1.0	1.25	.16259	.19515	.21984	.25481
1.0	1.75	.19896	.23728	.26362	.29897

TABLE VII

Total Frequency Errors (Hz)

Torque Tube I _y	Chordwise Modulus	Spanwise Modulus				
		0.75	1.25	1.75	2.25	2.75
.75	.75	17.5563	12.5356	9.4532	7.3145	5.7106
.75	1.25	14.8703	9.3196	5.8913	3.6134	2.2457
.75	1.75	13.2794	7.4264	3.8712	2.0229	2.4010
.875	.75	16.8579	11.9225	8.8956	6.7879	5.1994
.875	1.25	14.1054	8.6113	5.2075	2.9123	1.5574
.875	1.75	12.4813	6.6561	3.0698	1.2909	2.3115
1.0	.75	16.2305	11.3882	8.4340	6.3839	4.8457
1.0	1.25	13.4165	7.9991	4.6625	2.4460	1.3496
1.0	1.75	11.7602	5.9943	2.4618	1.1393	2.6339

TABLE VIII

First Torsion Frequency Errors (Hz)

Torque Tube I _y	Chordwise Modulus	Spanwise Modulus				
		0.75	1.25	1.75	2.25	2.75
.75	.75	5.4601	4.5120	3.8270	3.2910	2.8520
.75	1.25	4.0900	3.1930	2.5560	2.0560	1.6450
.75	1.75	3.2350	2.3770	1.7750	1.3040	0.9140
.875	.75	5.3060	4.3540	3.6600	3.1150	2.6690
.875	1.25	3.9200	3.0280	2.3870	1.8810	1.4650
.875	1.75	3.0560	2.2090	1.6050	1.1300	0.7370
1.0	.75	5.1710	4.2100	3.5040	2.9490	2.4940
1.0	1.25	3.7710	2.8780	2.2290	1.7150	1.2920
1.0	1.75	2.8980	2.0550	1.4460	0.9650	0.5660

Appendix C

Data Structure in PROCES

Modifications to PROCES. The version of PROCES used in this thesis had the same structure as that in Ref 2 with a few exceptions. The major difference is the type of optimization scheme used. Ref 2's PROCES used a linear technique while this version uses a non-linear routine called SUMT. Several modifications, all increases in capability, were made to make the program more general, as necessary for this thesis. They include an option to add to an existing NDSS, have a large number of members in the NDSS and compare X's before eliminating dominated solutions. No changes were made to SUMT except for an additional print option. A basic flow diagram of PROCES is shown in Figure 3 with the areas modified expanded in more detail. The SUMT box represents a series of subroutines making up about 90% of the actual program. All the blocks below SUMT are the one subroutine where the modifications were made.

The three modifications mentioned above were all made in the subroutine which constructs the non-dominated solution set from the SUMT solutions. All three changes were made as additional capability, not destroying any other capability. The first modification gives the user the option of continuing to build upon an existing NDSS. This is a useful option when the user must partition the performance space into smaller sections due to execution time limitations. In the T-38 example, the six-dimensional performance space was divided into 32

smaller spaces which could each be examined in one interactive terminal session. About 72 to 80 performance index combinations were investigated in each of the smaller subspaces. Each combination represents an optimization and a solution. The one session, then, produces a total of 72 candidate NDS's. This modification was made in the dashed area of Figure 3 labeled with a 1.

The next modification, represented by the dashed area and a 2, shows that the NDSS is stored in an external file rather than in internal arrays, allowing an almost unlimited number of NDSS members as opposed to the size limited arrays. This external file storage does pay the penalty, though, of being more time consuming. The arrays limited the NDSS size because of computer memory space limitations imposed on interactive programs. The NDSS members are recalled from the storage file one at a time to compare with the current solution. New NDSS members are then placed in the file as necessary.

The third area modified is also shown in Figure 3. Here, the X's of dominated NDSS members and the dominating solution are compared. Dominated members are kept if they are different. This modification checks for the condition in which the same performance set may be produced by different models in the decision variable space.

The last change was an additional print-out option for the entire program. This option leaves all other options intact but gives the added convenience of getting only the final solution for each optimization and the final NDSS when the program is finished, greatly reducing the volume of output data. This option is very useful for production runs of many subspaces.

Data Structure. The performance indices and constraint values for

a given \underline{X} are required in SUMT and also in the portion of the program depicted in the box immediately after SUMT (see Figure 3). A user supplied subroutine named RESTNT is called to obtain the objective function and constraint values. RESTNT contains the performance error 'functions' and constraint functions. A value is required for a single constraint or performance error index when RESTNT is called. The constraints, which are the table limits here, must also be the feasibility limits to the modelled problem. The constraints are calculated by single statement equations. The constraint value is equal to the decision variable value minus the appropriate limit.

In this finite element model case, the figures of merit are found by interpolating within data tables. RESTNT contains the data tables and the call to the interpolating routine. RESTNT supplies the appropriate data and \underline{X} to the interpolator for the performance error of interest and requires a performance error value in return. In turn, RESTNT computes the performance error value, the inequality constraint values and the performance constraint values and sends them back to the calling routine. Figure 4 shows how this system works in block diagram form and Figure 5 shows the version of RESTNT used in the stabilator example. Most of the data has been deleted from Fig 5 for clarity. IN is the index value indicating the particular constraint or performance error value that is requested and VAL is the value of the requested parameter. There are N tables, one for each performance. SPACINT is the three dimensional interpolation routine used.

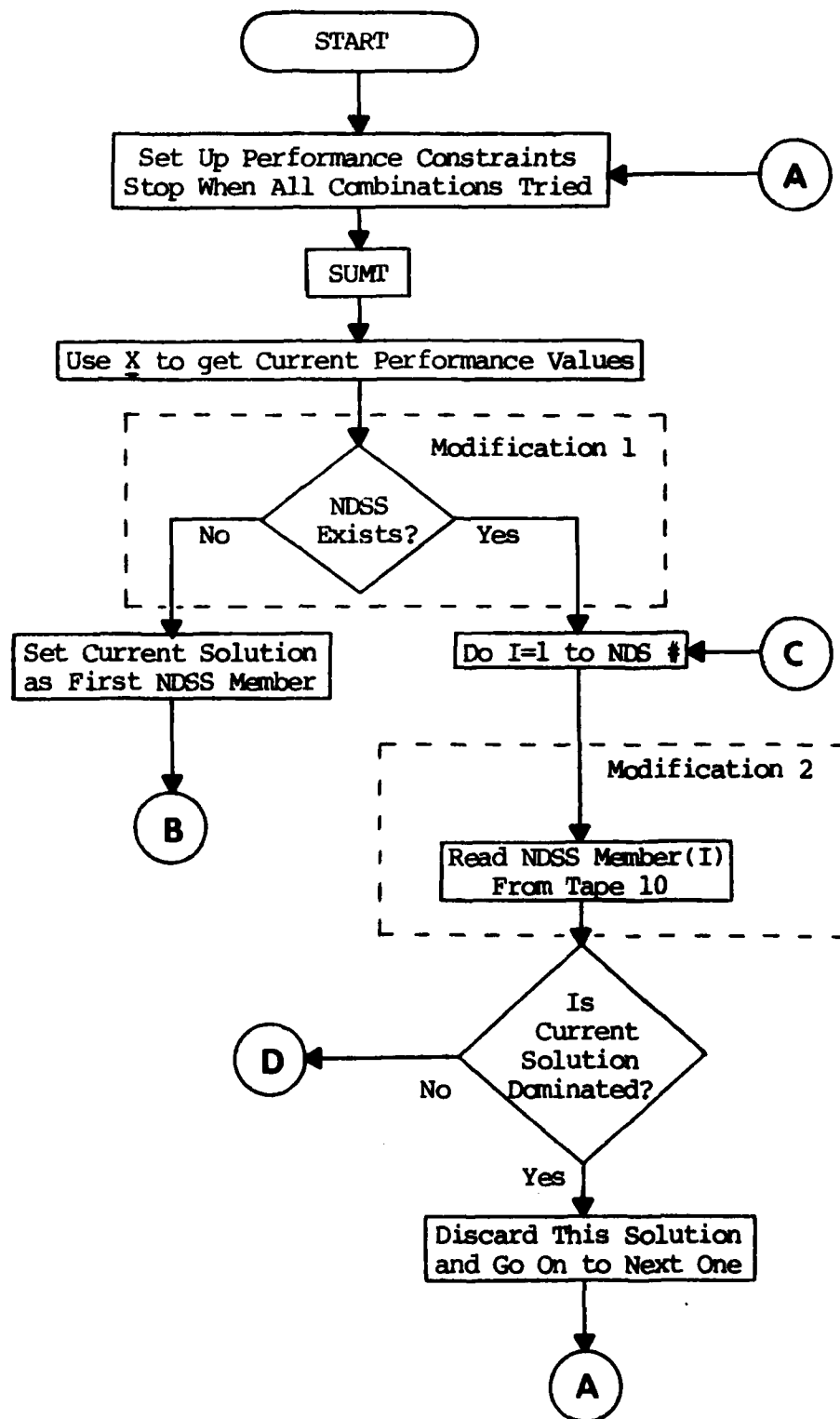


Figure 3A. PROCES Flow Diagram

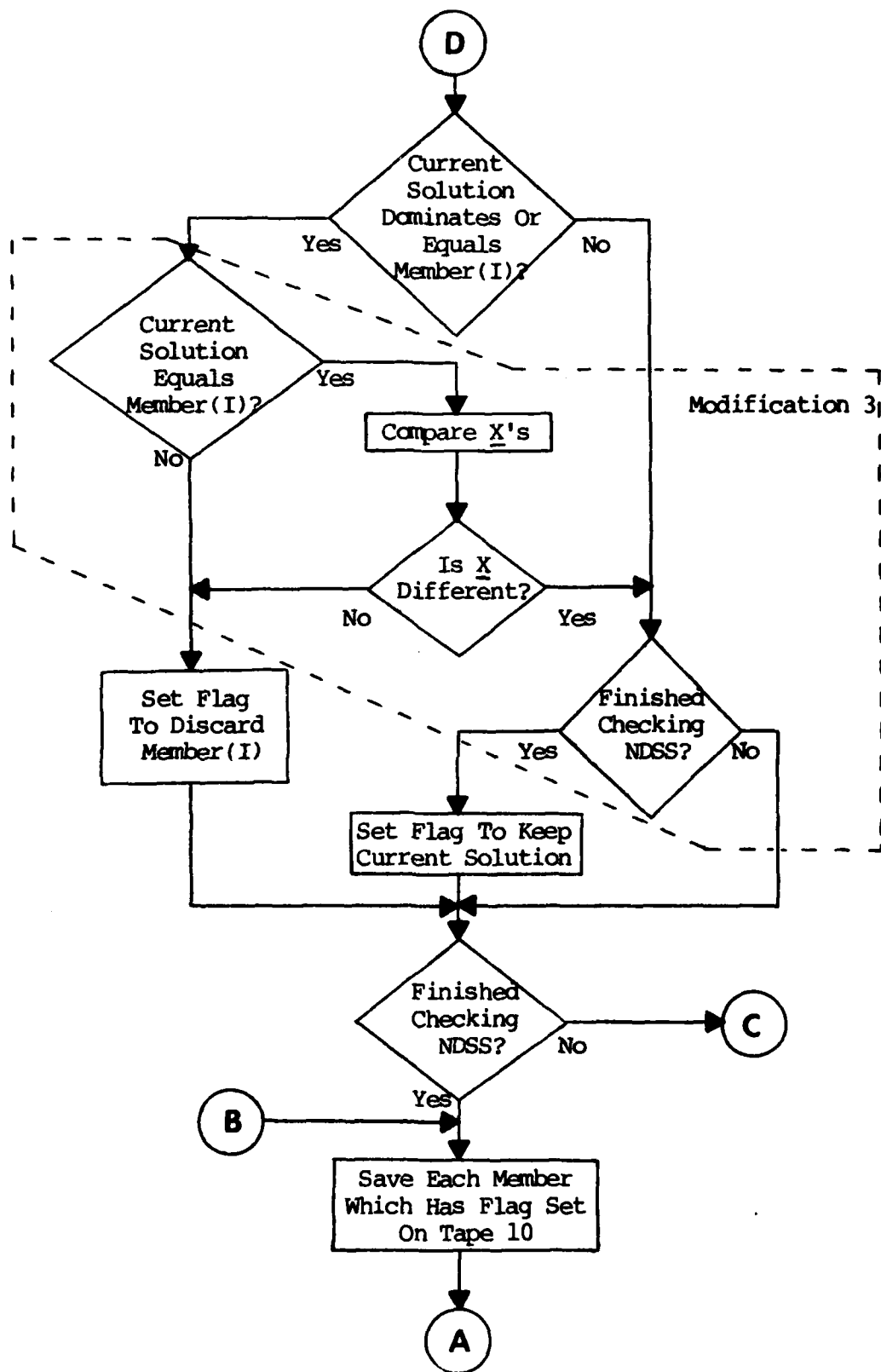


Figure 3B. Proces Flow Diagram

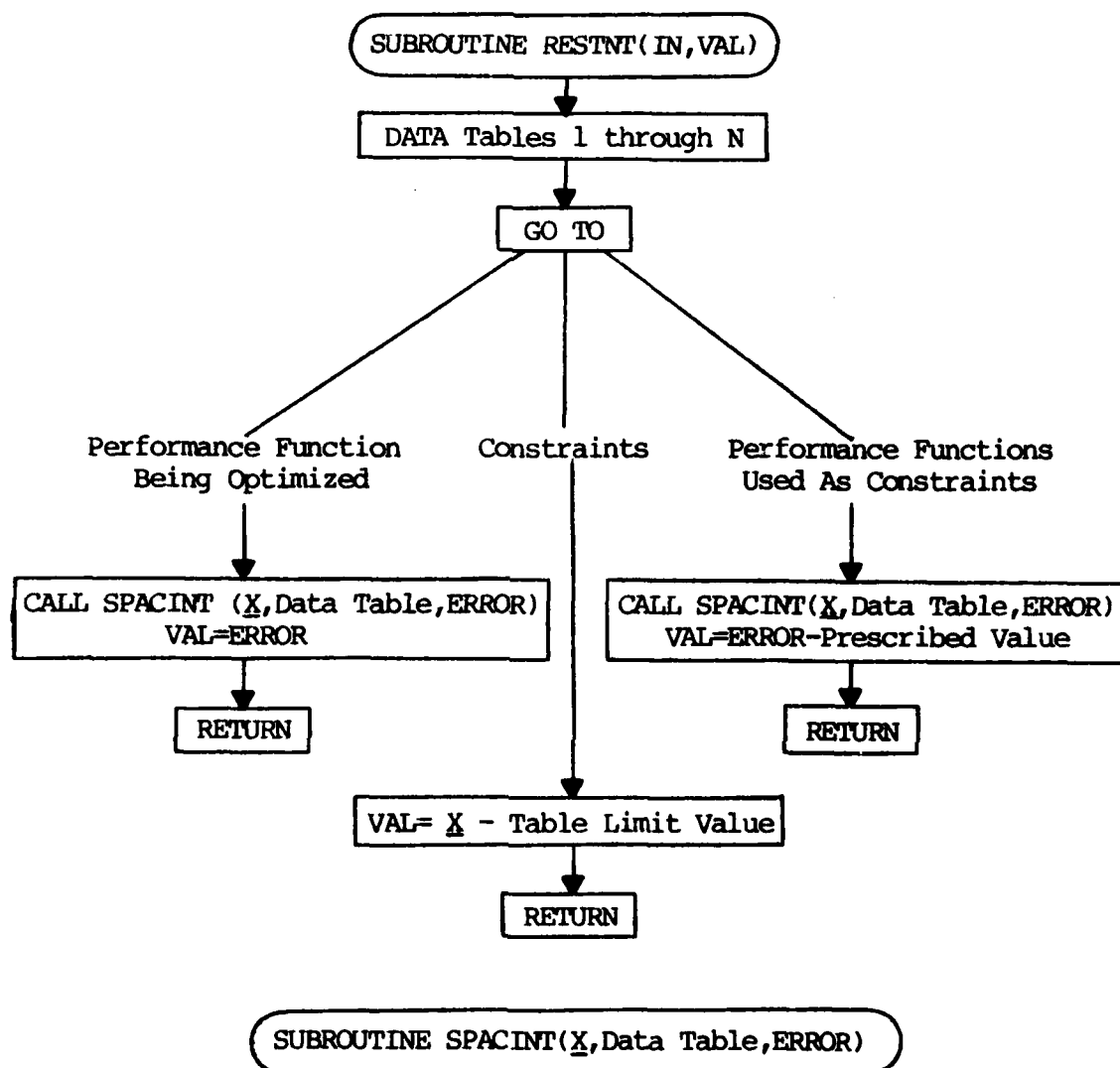


Figure 4. Data Structure in PROCES

```

SUBROUTINE RESTNT(IN,VAL)
COMMON/SHARE/X(25),DEL(25),A(25,25),N,M,MN,NP1,NM1
COMMON/ZB/ZZ(10),MZO,IT
DIMENSION ERR2T(4,3,4),ERR4T(4,3,4),
1   ERR10T(4,3,4),TTFTBL(3),
2   SEFTBL(5),ERRFT(3,3,5),ERRFTT(3,3,5)
DATA((ERR2T(1,I,J),J=1,4),I=1,3)/
1  4.2171, 1.98018, 2.389, 3.672,
2  3.62941, 1.78355, 2.48566, 3.902,
3  3.39106, 1.83167, 2.68496, 4.133/
DATA((ERR2T(2,I,J),J=1,4),I=1,3)/
1  3.66523, 1.83303, 2.62135, 4.070,
2  3.09467, 1.78127, 2.80924, 4.353,
3  2.79938, 1.93564, 3.05138, 4.600/
DATA((ERR2T(3,I,J),J=1,4),I=1,3)/
1  3.01901, 1.94288, 3.08316, 4.683,
2  2.50488, 2.07927, 3.36562, 5.013,
3  2.27460, 2.32108, 3.63332, 5.269/
DATA((ERR2T(4,I,J),J=1,4),I=1,3)/
1  2.36614, 2.49549, 3.89313, 5.618,
2  2.00231, 2.79605, 4.24626, 5.979,
3  1.89749, 3.08285, 4.52791, 6.240/
DATA((ERR4T(1,I,J),J=1,4),I=1,3)/
. . .

DATA((ERR10T(1,I,J),J=1,4),I=1,3)/
. . .

DATA((ERRFT(1,I,J),J=1,5),I=1,3)/
1  17.5563, 12.5356, 9.4532, 7.3145, 5.7106,
2  14.8703, 9.3196, 5.8913, 3.6134, 2.2457,
3  13.2794, 7.4264, 3.8712, 2.0229, 2.4010/
DATA((ERRFT(2,I,J),J=1,5),I=1,3)/ . . .
DATA((ERRFT(3,I,J),J=1,5),I=1,3)/ . . .
DATA((ERRFTT(1,I,J),J=1,5),I=1,3)/
. . .

DATA((ERRFTFB(1,I,J),J=1,5),I=1,3)/
. . .

DATA((ERRFTSR(1,I,J),J=1,5),I=1,3)/
. . .

```

Figure 5A. Subroutine RESTNT

```

DATA(TTTBL(I),I=1,4)/
1 .75, .80, .875, 1.0/
DATA(CETBL(I),I=1,3)/
1 .75, 1.25, 1.75/
DATA(SETBL(I),I=1,4)/
1 .75, 1.25, 1.75, 2.75/
DATA(TTFTBL(I),I=1,3)/
1 .75, .875, 1.0/
DATA(SEFTBL(I),I=1,5)/
1 .75, 1.25, 1.75, 2.25, 2.75/

C
C USE IF (IN) 1,1,3 FOR USING TOTAL STATIC ERROR AS Z1 AND
C DOING A STATIC OR STATIC AND FREQ. ANALYSIS
C USE IF (IN) 2,2,4 FOR USING TOTAL FREQ. ERROR AS Z1 AND
C DOING A FREQ. ANALYSIS ONLY
C
IF(IN) 1,1,3
C
1 CALL SPACINT(X(1),X(2),X(3),TOTERR,TTTBL,CETBL,SETBL,ERRTBL,
1 4,3,3)
VAL=TOTERR
RETURN
2 CALL SPACINT(X(1),X(2),X(3),FREQER,TTFTBL,CETBL,SEFTBL,ERRFT,
1 3,3,5)
VAL=FREQER
RETURN

C
C COME HERE FOR STATIC OR TOTAL ANALYSIS
C SET UP FOR: Z2 = CASE 2 STATIC ERROR
C Z3 = CASE 4 STATIC ERROR
C Z4 = CASE 10 STATIC ERROR
C Z5 = TOTAL FREQ. ERROR
C Z6 = FIRST TORSION ERROR
3 IM=IN-M+1
GO TO (10,15,20,25,30,35,40,45,60,65,70) IN

C
C COME HERE FOR FREQUENCY ANALYSIS ONLY
C SET UP FOR: Z2 = FIRST BENDING ERROR
C Z3 = FIRST TORSION ERROR
C Z4 = SECOND BENDING ERROR
4 IM=IN-M+1
GO TO (10,15,20,25,30,35,75,70,80) IN

```

Figure 5B. Subroutine RESTNT


```

C
C   HERE ARE THE TABLE LIMITS (INEQUALITY CONSTRAINTS)
10   VAL=X(1)-.75
      RETURN
15   VAL=1.0-X(1)
      RETURN
20   VAL=X(2)-.75
      RETURN
25   VAL=1.75-X(2)
      RETURN
30   VAL=X(3)-.75
      RETURN
35   VAL=2.75-X(3)
      RETURN
C
C   OBJECTIVE FUNCTION CONSTRAINTS (EQUALITY CONSTRAINTS)
40   CALL SPACINT(X(1),X(2),X(3),CAS2ER,TTTBL,CETBL,SETBL,ERR2T,
      1   4,3,3)
      VAL=CAS2ER-ZZ(IM)
      RETURN
45   CALL SPACINT(X(1),X(2),X(3),CAS4ER,TTTBL,CETBL,SETBL,ERR4T,
      1   4,3,3)
      VAL=CAS4ER-ZZ(IM)
      RETURN
60   CALL SPACINT(X(1),X(2),X(3),CAS1OR,TTTBL,CETBL,SETBL,ERR1OT,
      1   4,3,3)
      VAL=CAS1OR-ZZ(IM)
      RETURN
65   CALL SPACINT(X(1),X(2),X(3),FREQUER,TTFTBL,CETBL,SEFTBL,
      1   ERRFT,3,3,5)
      VAL=FREQUER-ZZ(IM)
      RETURN
70   CALL SPACINT(X(1),X(2),X(3),TORER,TTFTBL,CETBL,SEFTBL,
      1   ERRETT,3,3,5)
      VAL=TORER-ZZ(IM)
      RETURN
75   CALL SPACINT(X(1),X(2),X(3),FBERR,TTFTBL,CETBL,
      1   SEFTBL,ERFTFB,3,3,5)
      VAL=FBERR-ZZ(IM)
      RETURN
80   CALL SPACINT(X(1),X(2),X(3),SBERR,TTFTBL,CETBL,
      1   SEFTBL,ERFTSB,3,3,5)
      VAL=SBERR-ZZ(IM)
      RETURN
      END

```

Figure 5C. Subroutine RESTNT

Appendix D

Optimization Data for T-38 Stabilator Example

The tuning process was applied to the four static load cases only, the two frequency cases only, and two different starting points for the combined static and frequency case, giving four NDSS's. The combined case had five different solutions based upon the relative weighting between the static and frequency cases. The extremes of the five weighted answers were the same as those solutions for the static conditions only or the frequency conditions only.

Four NDSS's were built for this analysis. The case of only the four static load conditions minimized the total deflection error for each combination of cases 2, 4 or 10 deflection errors. Twenty non-dominated members were identified for this NDSS. Nine weighting combinations for the four performance indices were tried with one answer always being the best. Table IX lists this model. The frequency case minimized the total frequency error while using the first three natural frequency errors as constraints. This NDSS had 61 members. The 12 weighting combinations tried showed that one answer was always ranked first. Now combining the four static cases, the total frequency error and the 1st torsional mode error, a NDSS of 202 members was obtained using the static answers as a starting point and optimizing the total deflection error (see Figure 6). Thirty-seven weighting combinations indicated that only five different solutions were necessary to cover the whole weighting range. These five

solutions are actually seen in Figure 6 as the five clumps of solutions which have been circled. The midpoint of the clump is the model specified in Table IX. When the best frequency only model was used as a starting point, the NDSS had 255 members (see Figure 8). But, again, after 20 different weightings, only the same five solutions as before were evident. Figure 7 is a 2-D view of Figure 6. Comparing the coordinates for the non-dominated solutions of the two pictures, one can see how the solution procedure tracked along the edges of the infeasible regions, bunching up the solutions along these boundaries.

Therefore, of the 12 solutions obtained above only five unique solutions existed. These solutions depend on how all of the static cases are weighted against the two frequency errors. Also, the solutions for the combination cases with different starting points are almost the same so are averaged in Table IX. Thus, it can be seen that the five solutions of the averaged combined case contains all the other solutions. The pure static and pure frequency cases are in opposite corners of the variable space. Therefore, the static and frequency models are not the same and the combination of the two creates a compromise.

Sawdy's and Dodge's models are also listed in Table IX, as well as the performance values for all solutions. The PROCES tuned static model is very close to Sawdy's total displacement error. The PROCES tuned frequency model, though, is 66% better in total frequency error but 97% worse in first torsion frequency error.

Three models were examined by a flutter analysis procedure. These results are presented in Table X. The true flutter data is unknown but estimated values were determined by measurement and by comparisons with

known flutter data from other aircraft (Ref 15). These estimates are the target data of Table X. Dodge's model was in the low end of the flutter speed range but had very accurate frequency predictions. The PROCES tuned best frequency model and the best first torsion model predicted the flutter frequencies very well but were about 130 knots too slow in flutter speed. The PROCES tuned best static model was only 100 knots slow but was about 3 Hz low also.

TABLE IX
Best Models for the T-38 Stabilator

Condition Model	Static	Freq.	Combined					Sawdy	Dodge*
			85-100	60-80	25-55	20	0		
% Static	100	0					0	100	0
Torque Tube Iy	.79	1.0	.785	.82	.75	.86	1.0	.80	1.0
Chordwise Modulus	1.25	1.75	1.10	1.74	1.75	1.75	1.75	3.0	1.0
Spanwise Modulus	1.25	2.25	1.24	1.25	1.78	2.25	2.25	1.02	1.0
Total Static Error	1.782	—	1.845	2.043	2.711	4.849	5.976	1.6	—
Case 2 Error	.132	—	.135	.152	.255	.496	.587	—	—
Case 4 Error	.098	—	.108	.149	.253	.502	.592	—	—
Case 10 Error	.103	—	.123	.130	.117	.201	.290	—	—
Total Freq. Error	—	1.139	10.09	7.009	3.955	1.351	1.145	—	3.448
1st Torsion Error	—	.965	3.529	2.286	1.771	1.145	.963	—	.490

* Dodge used a 37% increase in thickness to chord ratio.

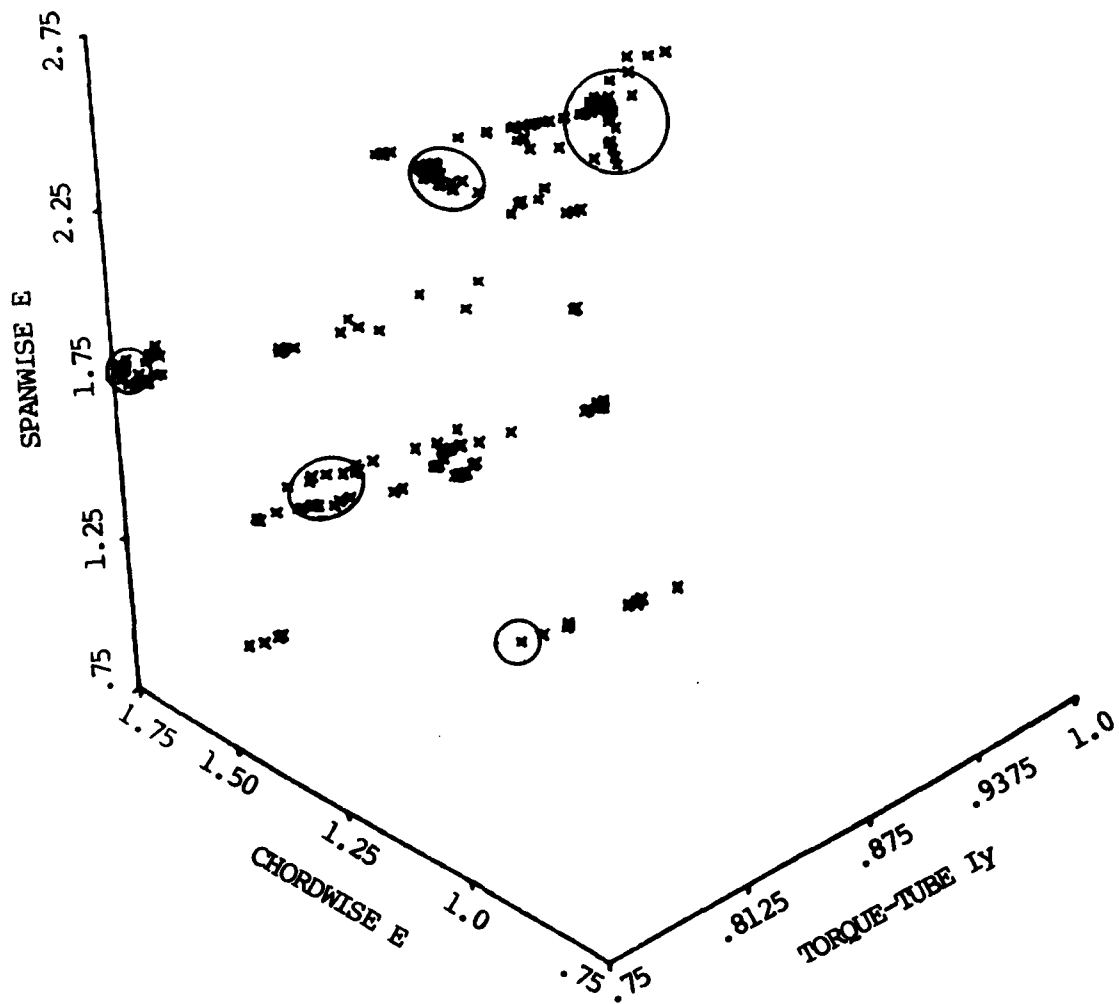


Figure 6. Variable Space for the 202 Member NDSS

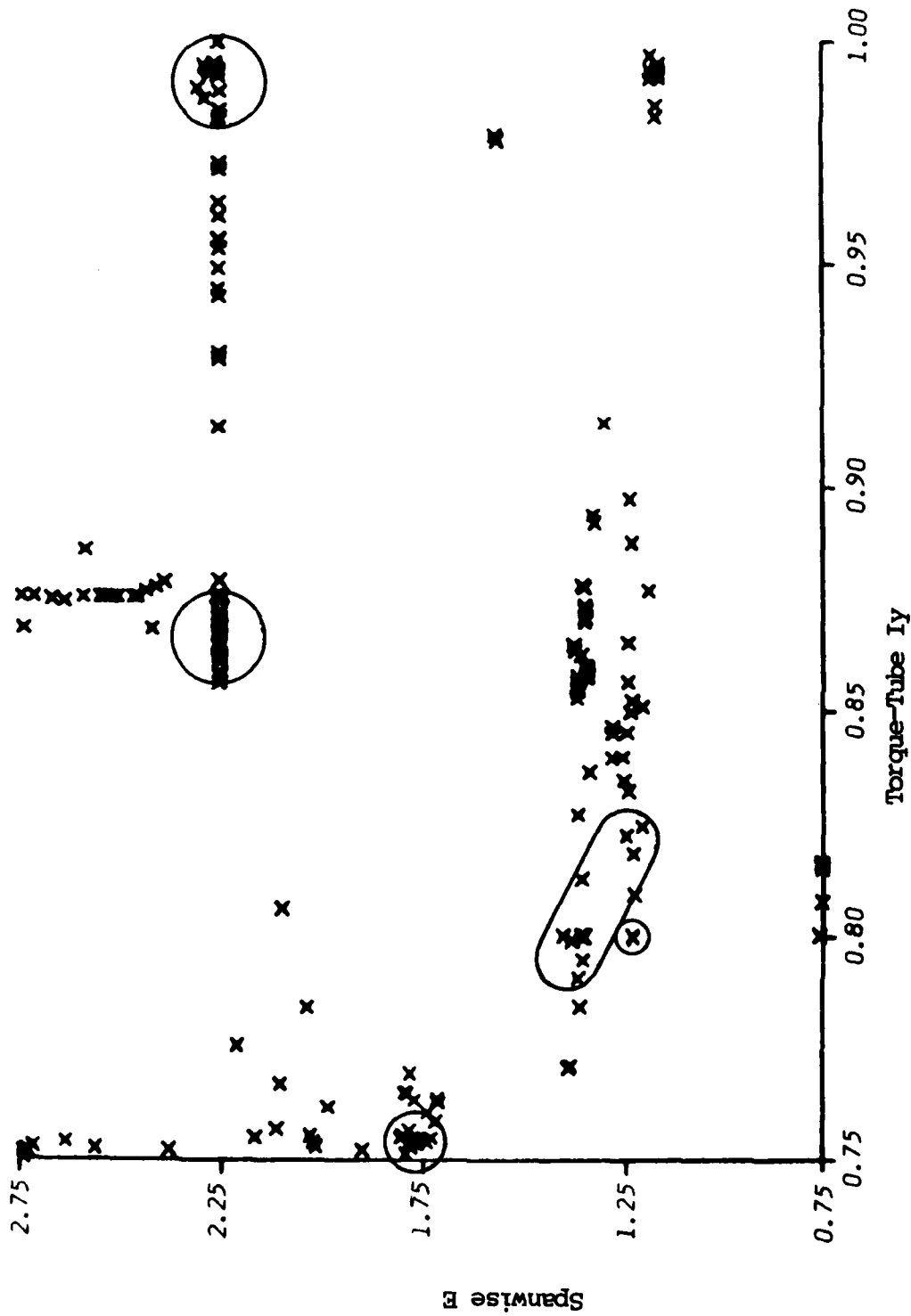


Figure 7. 2-D View of Variable Space for 202 Member NDSS

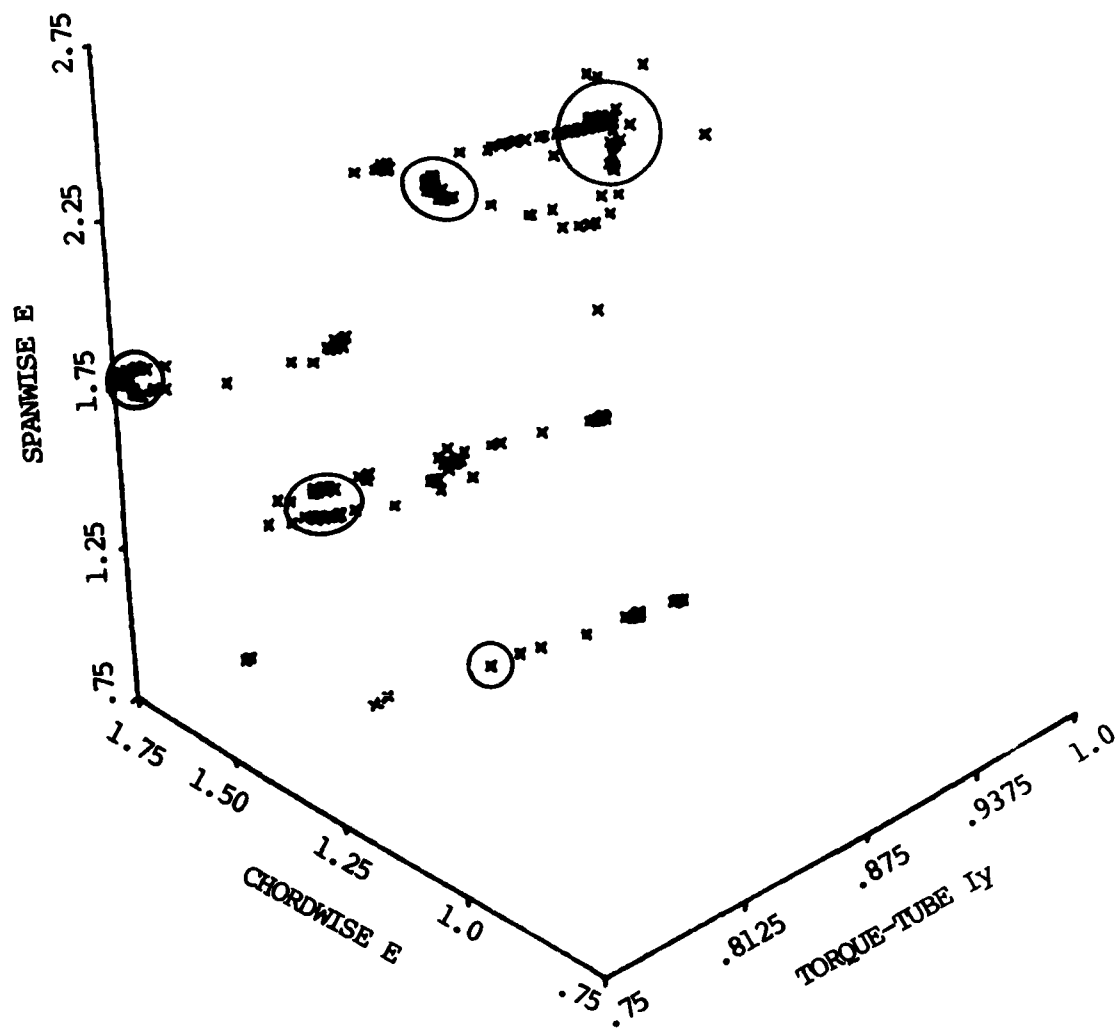


Figure 8. Variable Space for the 255 Member NDSS

TABLE X

Flutter Analysis Data

Model	Variables	Flutter Speed Knots	Flutter Freq. Hz	1st Torsion Freq. Hz
Target	- -	913-996	29.0	44.3
Dodge's†	1.0, 1.0, 1.0†	916.5	29.8	44.3
PROCES Static	.79, 1.1, 1.24	815.9	27.9	41.4
PROCES Freq.	1.0, 1.75, 2.25	786.4	29.8	43.9
1st Torsion	1.0, 1.75, 2.75	787.0	30.2	44.3

† Dodge used a 37% increase in thickness to chord ratio also.

‡ The variables are in the order of Torque-Tube I_y , Chordwise E and Spanwise E .

Vita

Charles Robert DeVore was born on 26 December 1951 in Canton, Illinois. He graduated from Easton Community High School in Easton, Illinois in 1970 and attended Western Illinois University until he enlisted in the USAF in 1971. He attended Ohio State University while on active duty and received the Bachelor of Science in Aeronautical and Astronautical Engineering degree in June 1977. He was commissioned through the USAF Officers Training School and assigned to the AF Rocket Propulsion Laboratory, Edwards AFB, California. While at Edwards, he served as a Rocket Plume Effects Analyst and later as an Air-to-Air Missile Propulsion Analyst and Program Manager. Capt DeVore was selected to attend the School of Engineering at the Air Force Institute of Technology, Wright-Patterson AFB, Ohio to earn a Masters Degree in Aeronautical Engineering in December 1982. His next assignment is with the Foreign Technology Division, Wright-Patterson AFB, Ohio.

Permanent Address: RRL, Scott's Trailer Court
Havana, Illinois 62644

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GAE/AA/82D-8	2. GOVT ACCESSION NO. AD A124791	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) STRUCTURAL MODEL TUNING VIA VECTOR OPTIMIZATION		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
7. AUTHOR(s) CHARLES R. DEVORE Capt USAF		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, OH 45433		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE December, 1982
		13. NUMBER OF PAGES 81
		15. SECURITY CLASS. (of this report) Unclassified
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES for public release: IAW AFR 190-17, LYNN E. WOLAVER Dean for Research and Professional Development Air Force Institute of Technology (AIC) Wright-Patterson AFB OH 45433 6 JAN 1983		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Model Tuning Vector Optimization NASTRAN Finite Element Model		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) <p>This report examines tuning a finite element model using vector optimization techniques. Structural models using finite element theory often need to be adjusted so they can accurately simulate the real structure. The goal is to tune the model such that it will reproduce data derived from the structure.</p> <p>The problem has been broken into two parts. First, the performance indices are extremized using multiple objective optimization theory, producing a set of possible solutions. Next, the solutions are rank ordered according to</p>		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

~~UNCLASSIFIED~~

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

a decision maker's preferences to select the best answer.

The tuning process was applied to a T-38 horizontal stabilator. Four static deformation and two natural frequency experimentally determined data sets were used as the objective functions for the three parameter model. This data was compared to the analytical data computed by NASTRAN, producing a set of over 200 models as the solution for the constrained, non-linear problem. Numerous weighting combinations indicated that only five of the solutions were of interest. These five solutions contained a best static deformation model, a best frequency model and three intermediate combinations of these two models.

The automated procedure outlined in this report proved to be a versatile method capable of solving many types of tuning problems.

~~UNCLASSIFIED~~

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

DATA((ERR1OT(1,I,J),J=1,4),I=1,3)/

....

DATA((ERRFT(1,I,J),J=1,5),I=1,3)/

1 17.5563, 12.5356, 9.4532, 7.3145, 5.7106,

2 14.8703, 9.3196, 5.8913, 3.6134, 2.2457,

3 13.2794, 7.4264, 3.8712, 2.0229, 2.4010/

DATA((ERRFT(2,I,J),J=1,5),I=1,3)/ . . .

DATA((ERRFT(3,I,J),J=1,5),I=1,3)/ . . .

DATA((ERRFTT(1,I,J),J=1,5),I=1,3)/

....

DATA((ERFTFR(1,I,J),J=1,5),I=1,3)/

....

DATA((ERFTSR(1,I,J),J=1,5),I=1,3)/

....

Figure 5A. Subroutine RESTNT

END

FILMED